ATOM NIEUWS

JAARGANG : 8

NUMMER : 3

DISKNRS. : 6-7



Bestuur:

Voorzitter:	Secretaris:	Penningmeester:
P. van Kuik	J.Hartog	T.Rutten
Zuideinde 54-a	Keyenbergseweg 60	Berkenlaan 24
1843 JF Groot-Schermer	6871 WK Renkum	3737 RN Groenekan
Tel.02997-1902	Tel.08373-13757	Tel.03461-3495

Clubwinkel:	Redactie Atom Nieuws:	Atomtel:
P.Grevelt	E.Sanders	
Emmastraat 22	Rosslag 13	databank van de
1782 PD Den Helder	6049 BE Herten	ATOM club
Tel. 02230-23453	Tel. 04750-30401	Tel. 020-167383

Contributie 1989 : fl. 25.00 : Atom Computer Club : Giro 5244293

Redactie Atom Nieuws		SPS-Printservice	Ledenadministratie
E.Sanders	04750-30401	E.Sanders	S.van Leeuwen
B.Tossaint	043-431675	Rosslag 13	Kompasstraat 32
N.Schreurs	04490-73679	6049 BE Herten	1973 PX IJmuiden
R.Leurs	04490-36454	04750-30401	Tel. 02550-22435
W.Truyen	09-3211564792	Zie blz. 21	

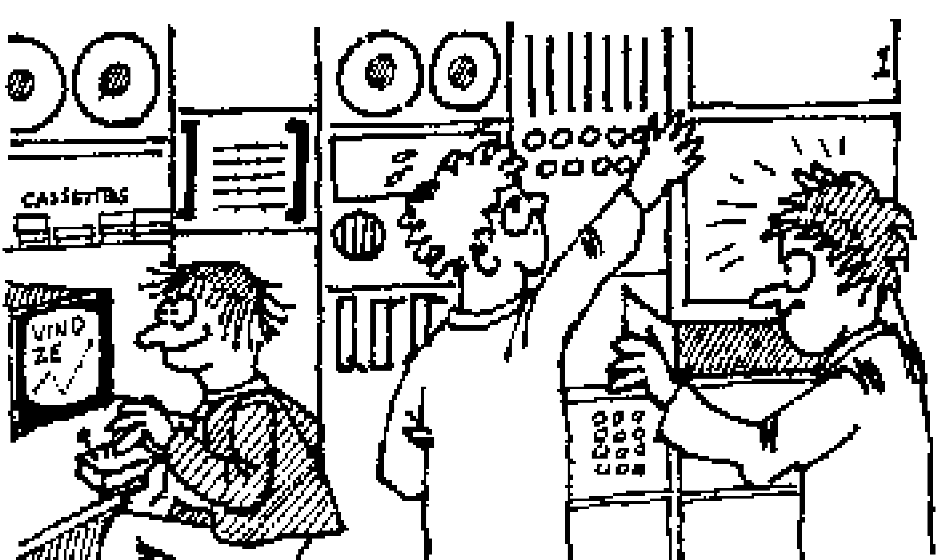
Uiterste datum inlevering KOPIJ: nr. 8-4   01 November 1989

De clubwinkel:

80-koloms videokaart	fl. 40.00
Geheugenkaart 16 kByte , exclusief onderdelen	fl. 1.00
(NEE, die fl. 1,-- is geen TYPfout!!!)	
Schakelkaart: deze is uitverkocht	
Minischakelkaart: ook deze kaart is uitverkocht	
Combikaart: Ze SPS Print-Service	
Z-80 kaart voor CP/M ,exclusief onderdelen	fl.100.00
ACORN NIEUWS 1982 , 97 pagina's samenvatting	fl. 2.50
ATOM NIEUWS jaargang 1983, ruim 450 pagina's	fl. 2.50
ATOM NIEUWS jaargang 1984, ruim 650 pagina's	fl. 2.50
ATOM NIEUWS jaargang 1985, ruim 650 pagina's	fl. 2.50
ATOM NIEUWS Jaargang 1986, ruim 500 pagina's	fl. 2.50
ATOM NIEUWS Jaargang 1987, ruim 300 pagina's	fl. 2.50
ATOM-WARE 1 : ATOM Basic interpreter , 98 pagina's	fl. 1.00
ATOM-WARE 2 : ATOM Disc Operating System , 68 pagina's	fl. 1.00
ATOM-WARE 3 : ATOM Monitor Operating System , 80 pagina's	fl. 1.00

LEVERING geschiedt via uw regionale penningmeester , of rechtstreeks via de federatieve penningmeester . Bij rechtstreekse bestelling stort u het bedrag van het gewenste artikel , vermeerderd met fl.4.00 portokosten , op de giro van de federatie onder vermelding van de naam van het artikel en uw lidmaatschapsnummer.

ATOM NIEUWS is een uitgave van de federatie Atom computerclub Ned/België en verschijnt 4 keer per jaar.
De redactie gaat er vanuit dat de ingezonden copy gemaakt is door de inzender tenzij in de publikatie uitdrukkelijk anders is vermeld. De aansprakelijkheid echter betreffende de auteursrech ten ligt zonder enig voorbehoud volledig bij de inzender.



Redaktie, ik zoek naar nieuwe kopij !

HET OPENINGSWOORD VAN DE REDACTIE

De Atom moet toch wel een echte goede computer zijn. Dit mogen we rustig zeggen omdat, ondanks dat het aantal pc gebruikers toeneemt en we een lange vakantie achter ons hebben liggen, we u toch weer een overvol Atom Nieuws kunnen presenteren.

We hebben ook ditmaal weer voor iedereen er wat erbij zitten. Zo vindt u in dit nummer o.a. de allernieuwste versie van de 80 kolomssoft: VDU3.4 van Peter Wokke. Deze soft had eigenlijk al in het vorige nummer moeten staan, maar dat was helaas misgelopen. Verder vindt u weer een deel van de cursus basic voor beginners, korte en lange programma's, een voorstel om de Atom te herzien en nog veel meer hardware, software en wetenswaardigheden.

Ook in dit nummer moeten we weer een oproep doen voor nieuwe kopy want het is ook nu weer op. En dat is eigenlijk zeer verwonderlyk want er zijn in onze club ongeveer 300 leden, en als van deze 300 leden er 100 zijn die regelmatig kopy insturen en er zijn 50 mensen die helemaal niets kunnen op hun Atom (gek eigenlijk want je kunt toch vanalles leren in Atom Nieuws) dan zijn er nog 150 leden die dus toch programma's maken maar die niets opsturen!

Zonder flauwekul verder, als u een programma schrijft, lang of kort maakt niets uit, en u bent te verlegen om zelf een artikel te schrijven dan kunt u gerust kontakt opnemen met de redactie, wij maken dan wel een passend stukje tekst. Het belangrijkste is dat we iets binnenkrijgen. Alle artikelen worden serieus behandeld !

Tot slot willen we de auteurs nog even vragen of ze bij hun diskettes een velletje papier doen met daarop de volgende informatie:
Filenaam - geheugen - utility - soort - artikel
Dit om het samenstellen van het overzicht van de regioschijven gemakkelijker te maken teneinde foute of ontbrekende files te voorkomen.

Wij wensen u veel plezier met uw en ons werk !!!

DE REDACTIE

ATOM NIEUWS REGIO-SOFTWARE '89

dnr	program	geheugen#	utility	soort	artikel	info
06	APS	2900-2ECC	—	UTILITY	AN.8-3	AUTOM.SAVEN OP DISK
06	CONVERS	3000-3520	P-CHARME	GRAPHICS	AN.8-3	ELECTRON PROGRAMMA
06	EDIT-SP	2900-2A58	—	UTILITY	AN.8-3	RANDOM-FILES
06	EXT-3.4	2900-5685	—	80 KOL.	AN.8-3	EXTENSIONS 3.4
06	FKILLER	2900-2A07	—	UTILITY	AN.8-3	FILE KILLER
06	FORMv32	2800-2E77	—	UTILITY	AN.8-3	FORMATTEER 32 KOL.
06	FORMv80	2800-2EFB	—	UTILITY	AN.8-3	FORMATTEER 80 KOL.
06	PICSHOW	3000-36CE	P-CHARM	GRAPHICS	AN.8-3	PLAATJES OP ELECTRON
06	SHACORN	2900-2BCA	TOBBOX	GRAPHICS	AN.8-3	BEGINN.LOGE
06	SURF1	2900-76BF	—	GRAPHICS	AN.8-3	DEMO VEKTOREN
06	SURF2	2900-65B4	—	GRAPHICS	AN.8-3	DEMO VEKTOREN
06	SURFING	8000-9800	—	GRAPHICS	AN.8-3	PLAATJE
06	UDS-ED	2900-3EDB	—	VDU	AN.8-3	80 KOL.SOFT
06	VDU-3.4	2900-5D66	—	VDU	AN.8-3	80 KOL.SOFT
06	VDU-UDS	2900-3EDA	—	VDU	AN.8-3	80 KOL.SOFT
06	VE-3.4	7400-7FAF	—	VDU	AN.8-3	80 KOL.SOFT
06	VECTORI	2900-2B6B	—	GRAPHICS	AN.8-3	OMZETTEN IN VECTOREN
06	VOORAAD	2900-2BF8	—	BESTAND	AN.8-3	BEGINN.LOGE
06	UDS-k.1	5000-53BF	—	VDU	AN.8-3	80 KOL.SOFT
06	UDS-k.2	5000-53BF	—	VDU	AN.8-3	80 KOL.SOFT
06	UDS-k.3	5000-53BF	—	VDU	AN.8-3	80 KOL.SOFT
07	BOOT-V2	2900-2EFD	P-CHARME	UTILITY	AN.8-3	MODEM UTILITY
07	CHQ.ASM	2900-3670	SALFAA	BESTAND	AN.8-3	CHEQUEBOEK
07	CHQ.BAS	8400-9FFF	P-CHARME	BESTAND	AN.8-3	CHEQUEBOEK
07	CHQBOOK	8000-9FFF	P-CHARME	BESTAND	AN.8-3	CHEQUEBOEK
07	CONVDAT	8200-8447	P-CHARME	UTILITY	AN.8-3	CHEQUEBOEK
07	D&L ASS	9100-995B	—	UTILITY	AN.8-3	MODEM UTILITY
07	D&L-FIL	3000-3D91	P-CHARME	UTILITY	AN.8-3	MODEM UTILITY
07	D&L-SRC	2900-49F3	SALFAA	UTILITY	AN.8-3	MODEM UTILITY
07	KAS.ASM	2900-366D	SALFAA	UTILITY	AN.8-3	KASBOEK
07	KAS.BAS	8400-9FFF	P-CHARME	BESTAND	AN.8-3	KASBOEK
07	REMOVSP	2900-2AFB	—	UTILITY	AN.8-3	INKRIMPEN VAN LISTINGS
07	RSCOM6	2900-40B2	SALFAA	COMMUN.	AN.8-3	MODEM UTILITY
07	SALF2.6	A000-AFFF	—	TOOLBOX	AN.8-3	SALFAA
07	V24-COM	2900-4140	SALFAA	UTILITY	AN.8-3	MODEM PROGRAMMA
07	VDT-V24	2900-4BC8	SALFAA	UTILITY	AN.8-3	MODEM PROGRAMMA
07	VDT-V25	9A00-9F49	—	MODEM	AN.8-3	MODEM PROGRAMMA
07	WORTELT	2900-2A0B	—	WISKUNDE	AN.8-3	WORTELTREKKEN

BEGINNERSLOGE VAN ROLAND LEURS.

```
BBBBB      AAAAA      SSSS      IIIIII     CCCC
BB  BB    AA    AA   SS  SS       II      CC  CC
BB  BB    AA    AA   SS           II      CC
BBBBB     AAAAAAA     SSSS        II      CC
BB  BB    AA    AA       SS       II      CC
BB  BB    AA    AA   SS  SS       II      CC  CC
BBBBB     AA    AA    SSSS      IIIIII     CCCC
```

deel 7

STRINGS IN ARRAY'S.

In deel 5 hebben we kennis gemaakt met arrays. In deel 6 leerden we werken met strings. In de vakantie hebben we natuurlijk veel geoefend. Dus zijn we nu op het moment aangekomen dat u leert om arrays met strings te gebruiken.
Verder komen in dit deel de uitbreidingsboxen aan de orde en ook behandel ik de werking van de cursur en copy toetsen. Er valt dus weer wat bij te leren...

Om meteen maar met een voorbeeld te beginnen: in deel vijf (bij de arrays) gebruikte ik het voorbeeld van een Acorn dealer. In dat voorbeeld kwam alleen aan de orde om de aantallen van een bepaald artikel op te slaan in een array. Maar nu we met strings werken kunnen we natuurlijk ook het artikel zelf in een array opslaan.
Het mooiste zou iets zijn in de geest van:
   DIM ARTIKEL$(100)
maar helaas werkt dit niet op een Atom. Wel gaat dit goed op bijv. een BBC, Electron of Atom met BBC-basic. Maar wij zullen het dus anders moeten doen.
Om nu duidelijk uit te leggen hoe dit alles in zijn werk gaat is het van belang om te weten wat we doen met het statement DIM.
Als we invoeren DIM A(65), dan krijgt de variabele A een waarde. Deze waarde is een geheugenplaats, meestal direct achter het basicprogramma. De DIM-pointer (= eerste vrije plaats voor een nieuwe string of array) wordt dan met 65 verhoogt. Met DIM PP(10) maken we elf arrayelementen aan. Dus als we strings in een array willen plaatsen moeten we er voor zorgen dat alle elementen PP naar een andere waarde A wijzen.

Dus eerst moeten we DIMmen voor het aantal array-elementen waar we strings in willen plaatsen. Vervolgens moeten we steeds weer opnieuw DIMmen om de plaats voor de strings zelf vrij te maken. Tot slot moeten we natuurlijk de inhoud van de array-elementen laten wijzen naar de string-plaatsen.
In een figuur ziet dat zo uit:

```
+-----------------------------------------------------------------+
: $#2A00 : $#2A10 : $#2A20 : $#2A30 : $#2A40 : $#2A50 : $#2A60 :
+-----------------------------------------------------------------+
    ^        ^        ^        ^        ^        ^        ^
+-----------------------------------------------------------------+
:  PP 0  :  PP 1  :  PP 2  :  PP 3  :  PP 4  :  PP 5  :  PP 6  :
+-----------------------------------------------------------------+
```

In basic zal het er dus zo uit moeten zien:

```
 DIM PP(29)
 FOR I=0 TO 29
   DIM A(33) ; PP(I) = A
 NEXT I
```

In de eerste regel maken we dus 30 array-elementen aan. In de tweede regel beginnen we een FOR/NEXT lus. De DIM-pointer staat nu dus achter de array PP. We maken dan voor $A 34 geheugenplaatsen vrij. De variabele A heeft nu een bepaalde waarde (bijv. #2A20). We maken dan PP(0) gelijk aan A, dus PP(0) wordt gelijk aan #2A20. De DIM-pointer staat op dit moment op A+33. Als we nu opnieuw DIMmen krijgt de variable A dus een nieuwe waarde. Deze waarde is 33 hoger dan de oude waarde. Dan wordt PP(1) gelijk gemaakt aan deze nieuwe waarde van A.
Op deze manier worden dus 30 strings gemaakt met een max. lengte van 34 karakters.

Onderstaand programma laat zien dat mijn theorie klopt:

```
 10 REM ADMINISTRATIE ATOM DEALER.
 20 REM ... beginnersloge deel 7 ...
 30 REM ROLAND LEURS - NATTENHOVEN
 40
 50 @=0 ; REM Veldbreedte voor variabelen op nul zetten.
 60 REM EERST DIMMEN EN ARRAYS MAKEN.
 70 H=5 ; REM AANTAL ARTIKELEN
 80 DIM AA(H),VV(H) ; REM AA=artikel-array, VV=voorraad-array
 90 FOR I=1 TO H
100   DIM S(64) ; AA(I)=S ; REM eerst dimmen en dan verwijzen
110 NEXT I
120
```

```
130 REM NU ARTIKELEN MET VOORRAAD INVOEREN
140 FOR I=1 TO H
150   PRINT "ARTIKEL "I" ; INPUT $AA(I)
160   PRINT "VOORRAAD "; INPUT A ; VV(I)=A
170 NEXT I
180
190 REM NU OVERZICHT PRINTEN.
200 PRINT "ARTIKEL :                    VOORRAAD :"'
210 PRINT "-----------------------------------"'
220 FOR I=1 TO H
230   PRINT $AA(I)
240   DO PRINT " ";UNTIL COUNT=26
250   PRINT VV(I)'
260 NEXT I
270
280 END
```

Zoals u in regel 150 kunt lezen is het toegestaan op strings regelrecht in een array in te lezen met het INPUT-statement. Dit kan omdat nu niet naar de inhoud van het array-element wordt gevraagd, maar naar de string die komt de staan op de plaats van de inhoud van het array-element.
Als onze dealer nu duizend artikelen wil opslaan in het geheugen, dan hoeft hij alleen maar de waarde van de variabele H te veranderen in regel 70.
Het programma kan nog worden uitgebreid met functies om artikelen te verwijderen, te wijzigen, te sorteren enz. In principe moet u er zelf een database van kunnen maken...

BASIC PROGRAMMA'S WIJZIGEN.

U heeft vast en zeker zelf wel eens een programma ingetikt. Daarbij zijn natuurlijk ook fouten gemaakt. In een paar alinea's wil ik u leren om te verbeteren en/of wijzigen m.b.v. de pijltjes links op uw toetsenbord en de COPY-toets.
Als u op een letter of cijfertoets drukt, dan komt dit overeenkomstig teken in de z.g. inputbuffer. Na een druk op de RETURN-toets verwerkt de computer de instrukties in de inputbuffer.
De COPY-toets plaats het karakter op de plaats van de cursor in de inputbuffer. De pijltjes besturen alleen de cursor. Zij veranderen de inputbuffer niet! Met dit in het achterhoofd kunnen we beginnen met het wijzigen. Stel we hebben deze regel ingetikt:

```
10 PRINY "ACORNATOM"';END
```

Als we nu de 'Y' in een 'T' willen wijzigen doen we het volgende:

-bestuur de cursor met de pijltjes tot voor de regel
-druk nu enkele malen op COPY, net zolang totdat de cursor op de 'Y' staat
druk nu op de letter 'T'
-druk nu net zo vaak op COPY totdat u de hele regel gehad heeft
-tot slot aan druk op RETURN

U kunt nu met LIST controleren of het goed gegaan is.
Stel nu dat we de tekst willen veranderen in "ACORN ATOM MICROCOMPUTER", uitgaande van dezelfde regel. Dit gaat als volgt:

-bestuur de cursor met de pijltjes tot voor de regel
-druk nu enkele malen op COPY, net zolang totdat de cursor voor het woord ATOM staat.
-ga nu met de pijltjes 1 positie naar links
-druk op de SPATIEBALK
-druk nu enkele malen op COPY, net zolang totdat de cursor voor het aanhalingsteken staat.
-ga met de pijltjes 1 regel naar boven
-typ nu in MICROCOMPUTER
-bestuur de cursor met de pijltjes tot op de aanhalingstekens
druk nu weer net zolang op COPY totdat u de hele regel gehad heeft
-eindig weer met RETURN

Het is hopelijk wel duidelijk dat als er een of meerdere letters verwijderd moet worden dat u met de COPY toets over de regel heengaat, en waar iets moet verdwijnen gebruikt u het horizontale pijltje. Probeer dat ook maar eens!

Zo ziet het werken met de cursortoetsen en met copy stap voor stap uit. De beste manier om dit goed te leren en te begrijpen is veel oefenen. Dus eerst veel fouten maken (heel gemakkelijk) en dan verbeteren (net zolang totdat dit ook gemakkelijk is!).

Als u een regel wil toevoegen, dan doet u dat gewoon. Dus als in uw programma staat :

```
10 REM SPACE QUEST II
20 REM DOOR P. VAN HEES
```

en u wil hier een regel tussenvoegen dan voert u deze regel in, het regelnummer moet dan liggen tussen 10 en 20, bijv. 12

```
12 REM een nieuw spel voor de atom
```

Met LIST ziet u dan:

```
10 REM SPACE QUEST II
12 REM een nieuw spel voor de atom
20 REM DOOR P. VAN HEES
```

Als u een regel wil verwijderen dan typt u alleen het regelnummer, gevolgd door een druk op de RETURN toets. De regel is dan verdwenen.

Als u regelmatig regels toevoegd en verwijderd dan blijft er van een mooie volgorde niet veel over. Hiervoor kunnen we het statement RENUM gebruiken, eventueel met twee getallen erachter. Het eerste getal is het eerste regelnummer, het tweede getal is **de stapgrootte. RENUM alleen is hetzelfde als RENUM** 10,10. **Dit statement komt voor in o.a. P-Charme en Joebox.**

UITBREIDINGEN IN DE COMPUTER.

We zijn aangekomen op het moment dat de belangrijkste en meest gebruikte functies, statements en combinaties behandeld zijn. Tenminste, wat het standaard Atom gebeuren betreft.
In het verleden heb ik wel vaker al eens gewezen op P-Charme en Joebox. **Maar** er is nog veel meer. Sommige uitbreidingsroms zijn **speciaal voor programma's geschreven, andere voor tekstverwerking, graphics, machinetaal** debuggers enz.

Er zijn dus vele roms gemaakt. Om deze **allemaal** uit te **leggen** zou **toch iets te ver voeren. En in de meeste gevallen is ergens een handleiding uit te halen. De meeste handleidingen geven** een duidelijk overzicht van de statements in de toolbox.
U kunt bij het drukwerk archief vragen voor deze handleidingen. **Mocht u er dan toch niet uitkomen, dan mag u mij gerust bellen.**

Een paar uitzonderingen **maak** ik voor enkele statements. Deze zal ik nu toelichten.

READ / DATA / RESTORE

We hebben gezien dat we op verschillende **manieren een waarde aan** een variabele kunnen toekennen. Een methode is met **het = teken, een andere manier was m.b.v. het INPUT statement.**
In enkele toolboxen kunnen we gebruik **maken** van het statement READ om een waarde aan een variabele toe te kennen. Een voorbeeld zal dit verduidelijken:

```
10 REM READ / DATA / RESTORE
20 RESTORE
30 FOR X=1 TO 10
40 READ Y
50 PRINT Y
60 NEXT X
70 END
80 DATA 10,9,8,7,6
90 DATA 5,4,3,2,1
```

In regel 20 wordt de **DATA pointer gezet** op de waarde achter het **eerste** DATA **statement** (hier dus in regel 80). In regel 30 wordt een FOR/NEXT lus begonnen. In regel 40 wordt **aan** Y de **waarde** toegekend die de DATA pointer **aangeeft**. Dus Y wordt 10. In regel 50 wordt deze waarde **afgedrukt. Regel** 60 laat het programma **teruggaan als X niet gelijk** is aan 10. Regel 70 eindigt het **programma.**

De regels 80 en 90 bevatten de DATA regels. Deze regels mogen overal staan in het programma. Er wordt niets uitgevoerd als het programma in een DATA regel komt.
De waarden achter een DATA statement mogen alle variabelen zijn die we kennen. Dus array elementen, strings, floating point getallen en zelfs functies :

```
10 RESTORE
20 Y=1;DIM T(20)
30 READ $T
40 READ %X
50 PRINT $T
60 FPRINT %X'
70 END
80 DATA "%X = ",COS Y
```

U ziet dat dus veel mogelijk is met READ en DATA. Een vuistregel is dat alles is toegestaan dat ook is toegestaan met het = teken. Een uitdrukking als X=7 kan ook geschreven worden als READ X;DATA 7.
Tot slot nog een opmerking m.b.t. RESTORE. Als we achter RESTORE een regelnummer zetten dan wijst de DATA pointer naar het eerste DATA statement dat vanaf dat regelnummer voorkomt.

SHAPE

Met het statement SHAPE uit de JOSBOX kunnen we een vastgelegd figuur op het scherm tekenen. Dit figuur leggen we vast in een tabel. Hoe deze tabel opgebouwd moet moet worden zal ik stap voor stap uitleggen, wat dit is het moeilijkste van het SHAPE statement. Het eerste byte van de tabel is de lengte van de tabel (in bytes). Zo'n tabel kan dus max. 512 tekeninstructies bevatten. De bytes die daarna volgen bevatten ieder twee instructies.

```
         +-------+-------+-------+-------+
Byte:    | par.1 | dir.1 | par.2 | dir.2 |
         +-------+-------+-------+-------+
Bits:       7-6     5-4     3-2     1-0
```

De eerste twee bits (7-6) bevatten de eerste plotparameter. Dit wil zeggen welke plotopdracht uitgevoerd moet worden. Dit kan zijn:

```
00 MOVE     de grafische cursor wordt verplaatst.
01 SET      teken een witte punt
10 INVERT   maak een witte punt zwart en een zwarte punt wit
11 UNSET    teken een zwarte punt
```

De daaropvolgende twee bits (5-4) bevatten de informatie in welke richting getekend moet worden.

```
00 LINKS
01 RECHTS
10 OMHOOG
11 OMLAAG
```

En hiermee is dus een instructie gemaakt. Van de 8 bits per byte hebben we er pas 4 gebruikt. Dus met de overige 4 bits kunnen we

hetzelfde doen. Bits 3-2 hebben dezelfde betekenis als bits 7-6 en de bits 1-0 hebben dezelfde betekenis als de bits 5-4.
Een voorbeeld: we ontwerpen een vierkantje....

parameter	richting	parameter	richting	
01	01	01	01	= #55
01	11	01	11	= #77
01	00	01	00	= #44
01	10	01	10	= #66

Uit deze tabel kunnen we ook de figuur bepalen: eerst twee stappen naar rechts, dan twee stappen naar beneden, dan twee stappen naar links en tot slot twee stappen naar boven : een vierkantje!

Als we zo'n regel met opdrachten hebben, kunnen we hieruit een binair getal van maken. Zo is de bovenste regel :01010101. Als we dit omrekenen krijgen we de hexadecimale waarde 55 uit. Voor dit omrekenen verwijs ik naar de tekst over getalstelsels in Atom Nieuws 7.2 blz 32. Ook kunt u onderstaande oneliner gebruiken voor het omrekenen van binair - hexadecimaal:

```
@=0;DOIN.$T.;A=0;F.I=0TOL.T.-1;A=A*2+(T.?I)&1;N.;P.&A';U.0
```

Als u zover bent kunt u de tabel samenstellen. Kies hiervoor een stuk vrij geheugen, bijv. #3000. Geef dit adres de lengte van de tabel. Hier moet u dus het aantal gebruikte bytes opgeven en niet het aantal instructies; in ons voorbeeld is de lengte 4.
Dus:

```
10 ?#3000-4
```

Dan volgen de instructies:

```
20 ?#3001=#55;?#3002=#77;?#3003=#44;?#3004=#66
```

En nu is de tabel klaar. De rest van het programma dient om de figuurtjes op het scherm te krijgen:

```
30 CLEAR 4
40 A=ABSRND %256
50 B=ABSRND %192
60 MOVE A,B  ;REM plaats waar figuurtje komt te staan
70 SHAPE #3000  ;REM beginadres shape-tabel
80 GOTO 40
```

Onderstaand programma heb ik gesloopt uit het programma "Acorn beeld" uit A.N. 4.6 . Hierin wordt het SHAPE statement gebruikt om het woord ACORN op het scherm te tekenen.

```
 10 REM SHAPE DEMO  BIJ BEGINNERSLOGE DEEL 7
 20 REM UIT: PROGRAM ACORN
 30 REM        A.H. 4-6
 40
 50 B=#3000
 60 B!0=#A951;B!2=#A9A9;B!4=#A9A9;B!6=#A9A9;B!8=#BBA9
 70 B!10=#BBBB;B!12=#22B0;;B!14=#1188;B!16=#1111;B!18=#1111
 80 B!20=#B8B8;B!22=#8A8A;B!24=#AAAA;B!26=#999A;B!28=#11B9
 90 B!30=#A919;B!32=#B999;B!34=#BBBB;B!36=#B8B8;B!38=#A888
100 B!40=#AAAA;B!42=#1122;B!44=#1111;B!46=#BB11;B!48=#BBBB
110 B!50=#11B1;B!52=#8A11;B!54=#8A8A;B!56=#A999;B!58=#8AAA
120 B!60=#8888;B!62=#1111;B!64=#1111;B!66=#1111;B!68=#BB9B
130 B!70=#BBBB;B!72=#AA8A;B!74=#8A8A;B!76=#A88A;B!78=#BBBB
140 B!80=#B3BB;B!82=#5204
150 CLEAR 4
160 FOR X=10 TO 256 STEP 50
170   FOR Y=0 TO 192 STEP 20
180     MOVE X,Y ; SHAPE B
190   NEXT Y
200 NEXT
210 END
```

Tot zover dan mijn verhaaltje over het statement SHAPE. Ik denk dat ik duidelijk genoeg ben geweest in mijn uitleg. Mocht dit niet het geval zijn, mag u gerust eens bellen of schrijven. Ik zal dan in een volgend deel dan op uw probleem terugkomen.

Voordat ik afsluit wil ik nog iets zeggen over de toekomstplannen wat betreft de beginnersloge. In het volgende nummer van Atomnieuws is er geen basic. Dan publiceer ik een minicursus tekstverwerking (die nu al klaar ligt in de brandkast). Daarin wil ik aan de hand van voorbeelden u de kneepjes van de tekst editor bijbrengen.
Daarna, dan is het al 1990, besteed ik twee of drie delen van mijn beginnersloge aan de GAGSROM. Deze rom is speciaal geschreven voor het maken van spellen. Eerst bespreek ik het graphics gedeelte van deze rom, daarna de sprites, en daarna wordt al onze kennis toegepast in een echt programma dat een beetje voorstelt...

Dus er valt nog genoeg te leren op uw Atom. Voorlopig kunt u er tegenaan met de theorie van deze aflevering. En nogmaals: als er iets niet duidelijk is mag u bellen of schrijven naar :

Roland Leurs
Nattenhoverkoestraat 5
6129 LH Nattenhoven
tel.: 04490 - 36454

Ik sluit af met mijn vriendelijke groeten...

```
***************************************
*   KAS - BOEK   Door Dick Protzman   *
***************************************
```

De programma's zijn verbeterd t.o.v. de oude versie.
Voor uitleg zie ACORN NIEUWS Jrg.3, nummer 6, blz. 19.
(Redaktie: voor die mensen die dat nummer niet meer hebben volgt op blz. 22 en 23 eeen overdruk van dat artikel.)
CHQBOOK en KASBOEK zijn ook met een (*) ervoor op te starten.
Voor de knutselaars onder ons zijn de assembler routines van de Engelse en de Nederlandse versies hierbij gevoegd.
Via het programma CONVDAT converteert men de oude data, aangemaakt met het oude gelijknamige programma KASBOEK, naar de nieuwe opzet.
Het programma is geschreven voor de 80 kolomskaart.

```
>L.
   10 PROGRAM KASBOEK
   20 REM BEDANKT BERRY LAM-HERSCHREVEN.DICK PROTZMAN
   30 REM P-CHARME/GDOS 1.5/FP.ROM/RAM #2000.#A000/80.KOLL
   40
   50 N=#28C0;O=#2880;J=#2870;V=#2860;W=#2820;Y=15;P=0;Q=0
   60 $J="uit";$V="scherm";%N.
   70 P.$12$15;GOS.z;HTAB52;P."    kasboek (c)dipro'88 v2"';GOS.z
   80aP.$30';F.X=1TO29;P." ";N.;P."menu                    "''$5'';HTAB20
   90 P."1.....NIEUW BESTAND MAKEN"';HTAB20
  100 P."2.....RECORDS BIJVOEGEN"';HTAB20
  110 P."3.....RECORDS VERANDEREN"';HTAB20
  120 P."4.....BESTAND WEGSCHRIJVEN,"';HTAB20
  130 P."5.....BESTAND LADEN VAN SCHIJF"';HTAB20
  140 P."6.....ZOEKEN EN NAAR ["$V"]"';HTAB20
  150 P."7.....BESTAND INFORMATIE"';HTAB20
  160 P."8.....ZET PRINTER [aan/uit]"';HTAB20
  170 P."9.....STOPPEN"'';M=0;?O=#D;GOS.t
  180 CASE Q OF
  190<1>P.'';GOS.v;IFQ=CH"N";G.a
  200bP.$30';HTAB25;P."nieuwe file maken"''''$5;HTAB10
  210 !#2900=#0000FF0D;!#2916=#291B;!#291A=0
  220 IN."DATUM VAN VANDAAG [ddmmjj] "$N';HTAB10;GOS.x;!#2902=A
  230 FIN."START BEDRAG VOOR SPAAR REKENING "%B;P.';HTAB10
  240 G=%(%B*1000)/10;!#2906=G;!#290A=G
  250 FIN."START BEDRAG VOOR BANK  REKENING "%B;P.''
  260 G=%(%B*1000)/10;!#290E=G;!#2912=G
  270 GOS.u;IFQ=CH"J";G.a
  280 G.b
  290<2>P.'';GOS.v;IFQ=CH"N";G.a
  300 P.$30';HTAB25;P."records bijvoegen"''''$5;HTAB10
  310 IN."DATUM VAN VANDAAG [ddmmjj] "$N
  320 K=0;IF?N=#D;A=!#2902;P.$11;HTAB38;GOS.w;G.d
  330 GOS.x;!#2902=A
  340dVT.7;HTAB10;P.$5
  350 XIFK=0;T=!#2916;P."[@] VOOR STOPPEN"'
  360 ELSE T=?#80+?#81*256;I=T-45;P.'
  370 IFT=#7FF1;VT.5;P.$5$7;HTAB27;P."geheugen vol";LI.#82AC;G.a
  380 A=!T;HTAB10;IN."DATUM OVERBOEKING [ddmmjj] "$N
```

```
390 IFK;IF?N=#D;P.$11;HTAB38;GOS.w;T=T+4;P.';G.a
400 IF$N="@";IFK=0;LI.#82AC;G.a;REM SORT
410 IF?N=#D;IF?O=#D;P.$7;G.d
420 IF?N=#D;P.$11;HTAB38;$N=$O;P.$N'
430 $O=$N;GOS.x;!T=A;T=T+4
440qS=0;P.';HTAB10;P."SPAAR OF BANK REKENING   [S/B] ?"
450 DO INK.Q;U.Q=CH"S"O.Q=CH"B";P.$Q'';HTAB10
460 IFQ=CH"B";S=S:2
470 P."BIJ OF AF VAN HET BEDRAG [B/A] ?"
480 DO INK.Q;U.Q=CH"B"O.Q=CH"A";P.$Q'';HTAB10
490 IFQ=CH"B";S=S:4
500 P."WILT U DIT MERKEN MET ** [J/N] ?"
510 GOS.t;IFQ=CH"J";S=S:1
520 ?T=S;T=T+1;P.'';HTAB10;XIFS&4;P."ONTVANGEN VAN :"
530 ELSE P."BETAALD AAN   :"
540 P." .......+.....+......................";HTAB24;IN.$N
550 IFK;IF?N=#D;P.$11;HTAB25;P.$T';T=T+36;G.c
560 G=LENN;DO N?G=#20;G=G+1;U.G>35;N?35=#D;$T=$N;T=T+36
580cP.';HTAB10;IN."BEDRAG          :"$N
590 IFK;IF?N=#D;P.$11;%B=!T/100;HTAB25;FIX\4,2%B;G.610
600 %B=VALN
610 G=%(%BX1000)/10;!T=G;T=T+4;P.'';GOS.u;IFQ=CH"N";G.d
620 IFK=0;!T=0;!#2916=T
630 IFS&4=0;G=-G
640 XIFS&2;!#2912=!#2912+G
650 ELSE !#290A=!#290A+G
660 IFK;LI.#82AC;G.a
670 G.d
680<3>P.$30';HTAB25;P."records veranderen"''''$5;HTAB20
690 P."1.....VERANDER DOOR EEN NAAM"';HTAB20
700 P."2.....VERANDER DOOR DE DATUM"'';HTAB20
710 P."3.....VERWIJDER DOOR EEN NAAM"';HTAB20
720 P."4.....VERWIJDER DOOR DE DATUM"'';HTAB20
730 P."5.....TERUG NAAR MENU"''
740 GOS.s;IFQ>5;G.740
750 IFQ=5;G.a
760 K=1;IFQ>2;K=0
770 P.''';HTAB20;P."SPAAR OF BANK REKENING [S/B] :"
780 DO INK.M;U.M=CH"S"O.M=CH"B";P.$M
790 R=0;Q=Q+10;IFM=CH"B";R=1
800 CASE Q OF
810<11>G.k
820<13>
830kP.'';HTAB20;IN."ZOEKEN NAAR WOORD :"$W
840 GOS.f;P.$5;LI.#8210;REM INIT
850qLI.#8227;REM SEARCH NAME
860 IF?#82;G.j
870 IFR;IFS&2=0;G.g
880 IFR=0;IFS&2;G.g
890 GOS.h;%C=0;GOS.r;IFM=CH"M";G.a
900 IFM=CH"N";VT.6;P.$4;G.g
910 IFS&4;E=-E
920 XIFS&2;!#2912=!#2912+E
930 ELSE !#290A=!#290A+E
940 IFK;GOS.f;G.d
950 GOS.y;G.a
```

```
960(12)G.m
970(14)
980mP.'';HTAB20;IN."ZOEKEN NAAR DATUM [ddmmJJ] :"$N
990 IF?N=#D;G.a
1000 GOS.x;!#80=A;!#8C=A;GOS.f;P.$5;LI.#8210
1020;LI.#8389;REM SEARCH DATE
1030 IF?#82;G.j
1040 IFR;IFS&2=0;G.i
1050 IFR=0;IFS&2;G.l
1060 GOS.h;%C=0;GOS.r;IFM=CH"M";G.a
1070 IFM=CH"N";VT.6;P.$4;G.l
1080 IFS&4;E=-E
1090 XIFS&2;!#2912=!#2912+E
1100 ELSE !#290A=!#290A+E
1110 IFK;GOS.f;G.d
1120 GOS.y;G.a
1130(4)P.$30';HTAB25;P."bestand wegschrijven"''''$5
1140 *CAT
1150 VT.20;HTAB20;P."onder welke naam :";IN.$N
1160 IF?N=#D;G.a
1170 %OSCLI"SA.","""",$N,"""","2900"," ","8000",$13
1180 G.a
1190(5)P.$30';HTAB25;P."bestand laden"''''$5
1200 *CAT
1210 VT.20;HTAB20;P."onder welke naam :";IN.$N
1220 IF?N=#D;G.a
1230 %OSCLI"LO.","""",$N,"""","2900",$13
1240 G.a
1250(6)P.$30';HTAB20;P."zoeken en naar "$V''''$5;HTAB20
1260 P."1.....SPAAR REKENING OP NAAM"';HTAB20
1270 P."2.....SPAAR REKENING OP DATUM"';HTAB20
1280 P."3.....ZOEKEN VAN DATUM TOT DATUM"'';HTAB20
1290 P."4.....BANK REKENING OP NAAM"';HTAB20
1300 P."5.....BANK REKENING OP DATUM"';HTAB20
1310 P."6.....ZOEKEN VAN DATUM TOT DATUM"'';HTAB20
1320 P."7.....TERUG NAAR MENU"''
1330 GOS.s;IFQ>7;G.1330
1340 IFQ=7;G.a
1350 R=0;IFQ>3;R=1
1360 Q=Q+20
1370 CASE Q OF
1380(21)G.o
1390(24)
1400oP.''';HTAB20;IN."ZOEKEN NAAR WOORD :"$W
1410 GOS.f;P.$5;LI.#8210
1420;LI.#8227
1430 IF?#82;G.j
1440 IFR;IFS&2=0;G.1
1450 IFR=0;IFS&2;G.1
1460 GOS.e;IFM=CH"N";G.a
1470 GOS.h;G.l
1480(22)G.n
1490(25)G.n
1500(23)G.n
1510(26)
1520nP.''';HTAB20;IN."ZOEKEN DATUM "
```

```
1530 IFQ=230.Q=26;P."VAN "
1540 P."[ddmmjj] :":IN.$N;IF?N=#D;G.a
1550 GOS.x;!#88=A
1560 XIFQ=230.Q=26;HTAB20;P."ZOEKEN DATUM TOT [ddmmjj] :";IN.$N
1570 GOS.x;!#8C=A
1580 ELSE !#8C=!#88
1590 IF?N=#D;!#8C=99999999
1600 IF!#88>!#8C;P.$7;G.a
1610 GOS.f;P.$5;LI.#8210
1620pLI.#8389
1630 IF?#82;G.J
1640 IFR;IFS&2=0;G.p
1650 IFR=0;IFS&2;G.p
1660 GOS.e;IFM=CH"N";G.a
1670 GOS.h;G.p
1680<7>P.$30';HTAB25;P."bestand informatie"''''$5;HTAB20
1690 P."LAATSTE DATUM GEBRUIKT : "
1700 A=!#2902;GOS.w;P.'';HTAB20
1710 P."EIND BEDRAG SPAAR REK. :";%B=!#290A/100;FIX\4,2\%B
1720 P.';HTAB20
1730 P."EIND BEDRAG BANK  REK. :";%B=!#2912/100;FIX\4,2\%B
1740 P.''';HTAB20
1750 LI.#8350;REM NO.OFF BLOK's
1760 P."AANTAL RECORDS GEBRUIKT : "E'';HTAB20
1770 P."AANTAL RECORDS NOG LEEG : "494-E'';HTAB20
1780 P."BESTAND VAN #2900 TOT #"&!#2916
1790jVT.22;HTAB17;P."druk op de [spatiebalk] voor menu :"
1800 DO INK.Q;U.Q=CH" ";G.a
1810<8>P.'';HTAB14;P."DE PRINTER IS [",$J,"] EN MOET ZIJN :";IN.$N
1820 XIF$N="AAN";$J="aan";$V="printer";P=1
1830 ELSE $J="uit";$V="scherm";P=0
1840 G.a
1850<9>P.$7$30';HTAB25;P."stop het programma"''''$5;HTAB20
1860 P."DENK AAN HET COPIEREN VAN UW"'';HTAB20
1870 P."BESTANDEN....?"''''
1880 GOS.v;IFQ=CH"N";G.a
1890 P.$12''"OK"';?#3FC=0;E.
1900eIFX<Y:X=X+1;R.
1910 X=1;VT.22;HTAB20;P."volgende [spatie=Ja/N=e] :"
1920 DO INK.M;U.M=CH" "O.M=CH"N";IFM=CH" ";VT.6;P.$5
1930 R.
1940fP.$30;IFP;IFQ>20A.Q<27;P.$2
1950 P.''"start bedrag :"
1960 XIFR;%B=!#290E/100;FIX\4,2\%B"  "
1970 ELSE %B=!#2906/100;FIX\4,2\%B"  "
1980 P."totaal bedrag :"
1990 XIFR;%B=!#2912/100;FIX\4,2\%B;IFP;P.$21''"BANK REKENING"$6
2000 ELSE %B=!#290A/100;FIX\4,2\%B;IFP;P.$21''"SPAAR REKENING"$6
2010 P.'';GOS.z;P." datum  merk     ontvangen of betaald aan"
2020 P."              erbij     eraf     totaal"'
2030 GOS.z;P.$3;%C=0;X=0;R.
2040hIFP;IFQ)20A.Q<27;P.$2
2050 P." ";GOS.w;P."  "
2060 XIFS&1;P."XX  "
2070 ELSE P."     "
```

```
2080 P.$0" ";%B=E/100
2090 XIFS&4;F;X\4,2\%B
2100 ELSE P."          ";FIX\4,2\%B;%B=-%B
2110 %C=%C+%B
2120 XIFS&4=0;P."  ";FIX\4,2\%C
2130 ELSE P."           ";FIX\4,2\%C
2140 P.'$3;R.
2150zF.X=1TO79;P."-";N.;P.';R.
2160yT=?#80+?#81X256;H=!#2916;T=T-45
2170 ICOPY T+45,H+4,T
2180 H=H-45;!H=0;!#2916=H;R.
2190xA=((N?0-48)*10)+(N?1-48)
2200 A=A+((N?2-48)*1000)+((N?3-48)*100)
2210 A=A+((N?4-48)*100000)+((N?5-48)*10000);R.
2220wSTR A,N;GOS.x;IFN?4=#30;P."0"
2230 P.A;R.
2240vHTAB20;P."bent u zeker [J/N] ?";G.t
2250uHTAB10;P."alles goed ingevuld [J/N] ?"
2260tDO INK.Q;U.Q=CH"J"O.Q=CH"N";P.$Q;R.
2270sHTAB20;P."maak uw keuze ? "$8
2280 DO INK.Q;Q=Q-48;U.Q>0A.Q<10;P.$Q+48;R.
2290rVT.22;HTAB20;P."is dit het record [J/N/Menu] : "$8
2300 DO INK.M;U.M=CH"J"O.M=CH"N"O.M=CH"M";P.$M;R.
```

```
>>L.
   10 PROGRAM CONVERT-DATA
   20 REM LOAD OLD DATA, THEN 'RUN' CONVERT-DATA
   25 O=#28C0;P.$12''
   30
   40 T=?#2916+?#2917*256
   50 T=T-1;!T=0;T=T+4
   60 REM T IS TOP OFF FILE
   70 ICOPY #2910,T,#291A
   80 P=#291A;?P=#FF;P=P+1;T=T+2
   90
  100aIF?(P+1)=0;G.z
  110 M=P;A=!P;P=P+4;S=?P;P=P+1
  120 $O=$P;P=P+L.P+1;E=!P;P=P+5
  130 N=45-(P-M)
  140 ICOPY P,T,P+N
  150 T=T+N;P=P+N
  160
  170 REM REWRITE VALUE'S
  180 O=LEN O;F.U=G TO 36;O?U=#20;N.
  190 O?35=#D;P.'A"      "$O"       ";%B=E/100;FIX\4,2\%B
  200 !M=A;M=M+4;S=S;Z;?M=S;M=M+1
  210 $M=$O;M=M+LEN M+1;!M=E
  220 G.a
  230
  240z !#2916=T-3;P.''
  250 *SAVE "DATA.CONVERT" 2900 8000
  260 END
```

```
>L.
   10\\   ASSEMBLER PROGRAMMA VAN
   20\\   KASBOEK    (c)DIPRO '88
   30
   40 IN."WAAR WILT U DE MC CODE :"A
   50 @=4;P.$12
   60 ASM-VERSION;P."IS BEZIG MET UW PROGRAMMA"''
   70 HTAB 5;P."PASS 1"';PASS 2;GOS.a
   80 HTAB 5;P."PASS 2"';PASS 1;GOS.a
   90 ASM-WARNING;P=P&#0000FFFF;P.'
  100 HTAB 5;P."CODE VAN #"&A" TOT #"&P'
  110 P.''"START ADRES INIT               :#"&init'
  120 P.''"START ADRES ZOEKDE'NAAM        :#"&search'name'
  130 P.''"START ADRES SORT'DATUMS        :#"&sort'
  140 P.''"START ADRES AANTAL'BLOKS       :#"&bloks'
  150 P.''"START ADRES ZOEKDE'DATUM       :#"&search'date'
  160 P.''"START ADRES XR.KASBOEK         :#"&ster'run
  170 P.'';END
  180
  190aASM-BEGIN
  200
  210.OPTION #40
  220.TABLE #7000,#8000
  230
  240\\ ZERO PAGE ADRES, ALLEEN VERANDEREN
  250\\ SAMEN MET HET HOOFD PROGRAMMA
  260:zp=#80
  270:statement=#300
  280.CODE A
  290
  300:init' LDA@#29;STA zp+1
  310 LDA@#18;STA zp;LDA@0;STA zp+2;RTS
  320
  330:end'search INY;LDA(zp),Y;BNE name'0
  340 LDA@#FF;STA zp+2;RTS
  350:search'name' LDY@0;LDA(zp),Y;BEQ end'search
  360:name'0 LDY@4;STY zp+3;LDX@#FF
  370:name'1 LDY zp+3
  380:name'2 INY;INX;LDA#2820,X;CMP@#D;BEQ fil'data
  390 CMP(zp),Y;BEQ name'2
  400 CPY@40;BEQ name'3
  410 INC zp+3;LDX@#FF;BNE name'1
  420:name'3 JSR calc'45;JMP search'name'
  430
  440:fil'data LDY@0;STY#385;STY#34F;STY#36A
  450 STY#373;STY#377
  460 LDA(zp),Y;STA#322;INY
  470 LDA(zp),Y;STA#33D;INY
  480 LDA(zp),Y;STA#358;INY;INY
  490 LDA(zp),Y;STA#334;LDX@#FF
  500:fil'1 INX;INY;LDA(zp),Y;STA#2800,X
  510 CMP@#D;BNE fil'1
  520 INY
  530 LDA(zp),Y;STA#326;INY
  540 LDA(zp),Y;STA#341;INY
  550 LDA(zp),Y;STA#35C
```

```
560 JMP calc'45
570
580:calc'45 LDA@45;CLC;ADC zp;STA zp
590 LDA zp+1;ADC@0;STA zp+1;RTS
600
610:sort' JSR#F7D1
620.A 30,10,10,10,10,10,10,10,10,"wacht",13,10,#EA
630 JSR bloks';LDA#326;STA#5A
640 LDA#341;STA#5B;LDX@0;JSR#F668
650:sort'0 LDA@0;STA#5C;STA#5D;JSR init'
660:sort'1 LDA zp;STA zp+3;LDA zp+1;STA zp+4
670 JSR calc'45;LDY@2
680:sort'2 LDA(zp),Y;CMP(zp+3),Y;BCC swap'str
690 BNE sort'3;DEY;BPL sort'2
700:sort'3 LDX@2;JSR#F671\\ VERHOOG 5C/5D
710 LDA#5C;CMP#5A;BNE sort'1
720 LDA#5D;CMP#5B;BNE sort'1
730 LDX@0;JSR#F668\\ VERLAAG 5A/5B
740 LDA#5A;BNE sort'0
750 LDA#5B;BNE sort'0
760 RTS
770
780:swap'str LDY@45;LDX@0
790:swap'0 LDA(zp+3),Y;STA#28C0,X
800 INY;INX;CPX@45;BNE swap'0
810 LDY@0;LDX@0
820:swap'1 LDA(zp+3),Y;STA#2800,X
830 INY;INX;CPX@45;BNE swap'1
840 LDY@0;LDX@0
850:swap'2 LDA#28C0,X;STA(zp+3),Y
860 INY;INX;CPX@45;BNE swap'2
870 LDY@45;LDX@0
880:swap'3 LDA#2800,X;STA(zp+3),Y
890 INY;INX;CPX@45;BNE swap'3
900 JMP sort'3
910
920:bloks' JSR init'
930 LDY@0;STY#326;STY#341
940 STY#35C;STY#377
950:blok'1 LDA(zp),Y;BEQ blok'3
960:blok'2 JSR calc'45
970 INC#326;LDA#326;BNE blok'1
980 INC#341;JMP blok'1
990:blok'3 INY;LDA(zp),Y;BEQ search'end
1000 LDY@0;BEQ blok'2
1010
1020:end'date INY;LDA(zp),Y;BNE date'0
1030 LDA@#FF;STA zp+2
1040:search'end RTS
1050:search'date' LDY@0;LDA(zp),Y;BEQ end'date
1060:date'0 LDY@2;LDX@2
1070:date'1 LDA(zp),Y;CMP#8B,X;BCC date'5
1080 BNE date'2
1090 DEY;DEX;BPL date'1
1100:date'2 LDY@2;LDX@2
1110:date'3 LDA#8C,X;CMP(zp),Y;BCC date'5
```

```
1120 BNE date'4
1130 DEY:DEX:BPL date'3
1140:date'4 JSR fil'data:RTS
1150:date'5 JSR calc'45:JMP search'date'
1160
1170:ster'run LDX#0
1180:ster'0 LDA ster'data,X:STA statement,X
1190 BEQ ster'1;INX:BNE ster'0
1200:ster'1 LDA@#03:STA#3FC
1210 LDA@#04:STA#12:LDX@#FF
1220 TXS:JMP#CE86
1230:ster'data
1240.B #FF,#E3,#C6 \\ CONTROLE BYTE'S VOOR P-CHARME
1250.B #46,#49,#58 \\ = FIX
1260.B #83 \\ #300+#80 START VAN STATEMENT LIJST
1270.B #89 \\ LOW BYTE
1280.B #00 \\ AFSLUITING LIJST
1290.B #4C,#23,#80 \\ = JMP#8023
1300.B #00 \\ STOP BYTE
1310
1320.END OF ASSEMBLY
1330 RETURN
```

S P S Sanders Print Service

Cassette interfaceprint	5.00
MDCR interface print	5.00
Battery backup printje	3.00
8K hoge geheugenprintje	5.00
#E000 naar #1000 voor oude schakelkaart	3.00
Omschakelprintje voor de 80K kaart	3.00
Bootstrapprintje de Moor	6.00
Voedingsprint MDCR 12V	5.00
Acoustische verbindings-tester	5.50
Combikaart	30.00
Viaprint Z80 (Atombus)	15.00
9e printerbit	7.00

Al deze printen zijn ook gebouwd en getest te bestellen tegen kostprijs onderdelen en een symbolisch bedrag. Even bellen graag. (Tel. 04750-30401)

Bestellen: op de clubavond te Sittard of door overmaking van het bedrag (+1,-- per print verpakkings- en verzendkosten) onder vermelding van de print(en) op gironr: 794739 tnv: E. Sanders, Hooslag 13 te Herten.

KAS-BOEK

Het programma "kasboek" houdt voor een giro- en een bankrekening de inkomsten en de uitgaven bij, waarbij tevens de saldi worden aangepast.

Het databestand wordt opgebouwd vanaf #2902.
De eerste 22 bytes bevatten de volgende informatie:
!#2902 laatste datum van wijziging.
Een datum wordt ingevoerd als JJMMDD en opgeslagen in 4 hex-bytes.
!#2906 beginsaldo bank
!#290A eindsaldo bank
!#290E beginsaldo giro
!#2912 eindsaldo giro
Bedragen worden als FP-variabele ingevoerd, met 100 vermenigvuldigd en als integer-variabele opgeslagen en verwerkt.
?#2916 en ?#2917 top van het bestand.

Elk datablok bestaat uit:
datum (4 hex-bytes)
selectiebyte: afschrift ontvangen: JA -> bit0=1
bank of giro: BANK -> bit1=1
bij of af: BIJ -> bit2=1
bit3 t/m bit7 is vrij
naam, afgesloten met #0D
bedrag (4 hex-bytes integer)
scheidingsteken (#AF).
Na het laatste datablok komt als afsluiting een "lees" blok (zie PROC TAIL(U)) dat nodig is bij zoeken en sorteren.

Met behulp van menu's kunnen de verschillende functies van het programma worden gekozen.

1. "EERSTE INVOER". Hiermee worden de beginwaarden ingevoerd in de eerste 22 bytes.

2. "TOEVOEGEN AAN BESTAND" geeft vervolgens de mogelijkheid regelmatig nieuwe gegevens toe te voegen. Deze gegevens worden op datum gesorteerd.

3. "EDITEN" biedt de mogelijkheid wijzigingen in het bestand aan te brengen.
Met "gegevens wijzigen" kunnen veranderingen worden aangebracht in datum, selectiebyte en bedrag. Fouten in de naam kunnen hiermee niet worden hersteld. Het blok met de foutieve naam moet worden verwijderd en opnieuw ingevoerd.
Met "datablok verwijderen" wordt een bepaald blok uit het bestand gehaald en het eindsaldo wordt gewijzigd.
Met "t/m bepaalde datum verwijderen" kan een bestand dat te groot wordt, worden ingekort. Vanaf het begin tot de gekozen datum

worden alle gegevens verwijderd en het beginsaldo wordt bijgewerkt.

4&5. "BESTAND NAAR/VAN DISKETTE" zorgt voor het opslaan en weer terughalen van gegevens op/van de schijf. Om ongelukken te voorkomen wordt gevraagd of je het zeker weet, waarop met het hele woord JA of NEE moet worden geantwoord.
Bij het gebruik van cassettes i.p.v. een schijf zijn de volgende wijzigingen in het programma nodig.

```
regel   40 ?#FE=#7C:COS1
   (DISK schakelt de DOS in (JSR#C4E4:JSR#E000:JMP#C55B)
regel 1780 verwijderen
regel 1830 IF$C="JA":G.1870
regel 1860 verwijderen
```

6. "AFDRUKKEN" zet de gegevens op papier.
Er kan worden gekozen tussen of bank, of giro, of beide.
Daarna kan worden gekozen tussen selectie op naam, selectie op datum of geen selectie.
Bij het zoeken op naam hoeft niet de hele naam ingevoerd te worden. Het ingevoerde begin van de naam wordt tot #0D (CR) vergeleken met het begin van de naam in het bestand.
Bij het zoeken op datum kan tot of vanaf een bepaalde datum worden gezocht of tussen twee data.

7. "BESTANDS-INFORMATIE" toont het aantal data-blokken en het geheugengebied dat het bestand in beslag neemt.

In zijn volledige vorm is het programma te groot om vanaf #8200 te laden.
Wanneer echter de machinetaalroutines in geassembleerde vorm achter een ingekort programma komen, kan het wel.
Dit is op de volgende wijze te realiseren:
* Laad het hele programma vanaf #5800 en RUN het.
* (escape)
* Vraag de werkelijke adressen op van LL20, LL0, LL9, LL16, LL13, LL31, LL38 en LL3.
* Vervang deze labels in de PROCEDURES door de adressen.
* Verwijder in regel 60 LL41.
* Verwijder de regels 70 en 590.
* Verwijder de regels 3010 en verder.
* Bepaal TOP en verhuis het programma met COPY naar #8200.
* *SAVE"KASBOEK"8200 A000 C2B6

VAN JVDU 0.0 NAAR 0.1

De JVDU 0.0 soft kenmerkt zich door zijn eenvoud, compactheid en snelheid. Daarbij is de hoge snelheid de meest opvallende eigenschap. In dit artikel leest u hoe het nog sneller kan en verder enkele aanwijzingen om de soft naar eigen smaak aan te passen.

CLEAR-LINE ROUTINE

Bij het afdrukken van een blokgraphic karakter op de allerlaatste beeldschermlocatie in 40 kolom mode, blijkt dat de clear-line routine niet "waterdicht" is. Dit euvel kan simpel opgelost worden door tijdens de clear-line routine het atribuut byte B tijdelijk op 0 te zetten. Nu we toch aan het veranderen zijn, kunnen we meteen nog een andere wijziging doorvoeren. De routine kan namelijk nog een stuk sneller door gebruik te maken van het MVT commando. Breng daartoe de volgende wijzigingen aan:

```
1075:mvt = #F0 \ move triple buffer
1112:mvam= #08 \ move from auxpointer to mainpointer
1114:stop= #01 \ stop move at end of line

2515:print'sp  \ print space

3410 LDY bytB              \ save bytB in Y-register
3415 LDA @0
3420 STA bytB              \ clear graphic40 attribute
3425 STA axptr             \ set X-part of auxpointer at leftmost position
3430 JSR print'sp          \ print space at position 0 (even)
3435 JSR print'sp          \ print space at position 1 (odd)
3440 STY bytB              \ restore graphic40 attribute
3445 LDA myptr             \ get Y-part of mainpointer
3450 STA ayptr             \ set Y-part of auxpointer on the same line
3455 LDA @mvt+mvam+stop    \ move triple from aptr to mptr for one line
3460 BNE exec'cmd          \ execute command (branch always)

3555:rts2
```

Om de lengte van de soft binnen 3 pagina's te houden kunnen de regels 3910 t/m 4000 verwijderd worden. Ze zijn volkomen overbodig.

AUTO-REPEAT ROUTINE

Velen vinden de repeteer-snelheid bij auto-repeat te laag. Dit is eenvoudig te verhelpen door op regel 5650 de DEX instructie te vervangen door LDX @2. Er is dan nog steeds verschil tussen de snelheid van auto-repeat en de repeat-toets maar wel veel minder. Desgewenst kan men ook de snelheid van de repeat-toets verhogen door regel 5560 te veranderen in LDX @1. Verder is de aanspreektijd van auto-repeat met de waarde op regel 5540 in te stellen.

INTENSITEIT

De EF9345 heeft geen voorziening om de intensiteit van het beeldscherm in te stellen. Op een monochroom beeldscherm kan men verschil in intensiteit simuleren door de voorgrondkleur te veranderen. Wie de intensiteit van wit te hoog vindt, kan regel 1020 veranderen in :fgnd=#06 \ cyaan.

CURSOR

Als men voor een lagere intensiteit kiest door de voorgrondkleur op 6 in te stellen, valt het op dat de intensiteit van het cursorblokje niet mee veranderd. Het enigste wat men daar aan kan doen is het blokje te veranderen in het underscore teken. Op regel 1010 wordt dit ingesteld: underscore/vast = #60, underscore/knipperend = #70, blok/knipperend = #50.

PAGE MODE

Normaal staat de page mode uit na het opstarten van de vdu-soft. Vanwege de grote scroll-snelheid zal de page mode in de praktijk vrijwel altijd aan staan. Het is dan ook veel handiger om de page mode meteen bij het opstarten aan te zetten. Dit is eenvoudig te realiseren door toevoeging van de volgende regel: [605 JSR pgon'.

HOGERE CPU-KLOKFREQUENTIE

Er zijn geruchten dat de JVDU soft niet goed zou functioneren bij hogere klokfrequnties (1.8 of 2 MHz). Dat ligt volgens mij niet aan de software want bij mij draait de soft zonder problemen op 2 MHz. Ik heb de timing op de vdu-kaart aangepast door C1 en C2 te verlagen tot resp. 3p3 en 18 pF. Verder is het aan te raden om de 6502 te vervangen door een 2 MHz type.

JVDU VERSUS VDU-3.X

In een vorig artikel heb ik mij nogal negatief uitgelaten over de huidige vdu-software. Met de manier waarop ik dat gedaan heb, ben ik achteraf gezien toch niet zo gelukkig. Mijn verontschuldigingen daarvoor.
Of die kritiek nu terecht was of niet, feit is dat door dat artikel toch wat meer mensen over de vdu materie zijn gaan nadenken. Over een eventueel vervolg naar JVDU 1.0 lopen de meningen nogal uiteen. De 3.X soft van Peter Nokke is vooral bij BASIC georienteerde programmeurs populair. Gezien de vele extra's kan ik me dat ook best goed voorstellen. Deze groep ziet weinig heil in weer een nieuwe soft. De meer op assembler gerichte programmeurs daarentegen onderkennen veelal het tekort aan algemene besturingscodes. Zij vinden in 't algemeen een vervolg de moeite waard. Geen al te brede basis dus, maar het biedt toch enig perspectief om tot een meer algemeen bruikbare controlcode-set te komen.



"LAAT ONS DE ATOM HERZIEN."

B.Tossaint.

Na bijna 10 jaar sleutelen aan de ATOM op allerlei manieren, is de verzameling aan kaarten,uitbreidings-versies e.d. zo groot geworden,dat het al bijna op een ondoorzichtig oerwoud begint te lijken .
Oude rotten in het vak ,voorzover actief ,weten dat nog wel redelijk uit elkaar te houden,gewone gebruikers,en zeker nieuwelingen,-ook die zijn er inderdaad toch nog-,raken gauw het spoor bijster, danwel,en dat is erger ,krijgen de zaak niet meer aan de praat.
Ook waren er in de loop van de tijd steeds weer opnieuw moeilijkheden met uitbreidingen,rack-systemen,e.d.

Veel van die problemen hebben te maken met:

- # het niet op de juiste wijze manouvreren met de signalen "binnen en buiten de bus";
  bij vele wijzigingen,of aanvullingen moeten steeds weer allerlei extra draadjes gelegd worden om de "bus -schakeling " op de juiste wijze te bedienen.
- # toch onvoldoende sterkte (buffering) van stuursignalen

Daarnaast beginnen diverse gebruikers de ATOM opnieuw te bouwen; ofwel, omdat het oorspronkelijke moederbord teveel losse contacten en dus kuren begint te vertonen;
ofwel, omdat men de ATOM nog eens "echt goed " wil herbouwen.

Veranderen,vernieuwen,etc is aardig,maar je moet eigenlijk eerst bezien hoe de ATOM vandaag in elkaar steekt.
Daartoe is het ATOM schema ,met de meest voorkomende uitbreidingen heel schematisch opgezet,waarbij speciale aandacht is geschonken aan de weergave van de "bussen" en de verschillende bufferingen ,die,dat weet men ,zowel goed doen als ook verdragen.
Zie schema 1 en 2.

Bekijkt men de aaneenschakeling c.q.serie schakeling van adres en data lijnen via de diverse buffers,danwel de rechtstreeks koppeling, dan is het eigenlijk een wonder dat alles nog enigzins behoorlijk functioneert; maar is ook duidelijk ,dat er maar weinig fout behoeft te gaan ,en de zaak loopt vast

Een eerste stap op de weg der verbetering,door diverse mensen reeds gezet,-doorgaans echter in wire-wrap uitvoering,-is het bufferen van alle signalen die van de 6502 afkomen ,zoals in feite indertijd ook door ELEKTUUR werd gedaan .
Daarna kan "binnen de bus" en "buiten de bus" worden kortgesloten, zoals eerder al eens werd voorgesteld.

Er onstaat dan globaal het volgende schema:

```
6502 -- buffers          BUS
      :  v.
      :  ADRES       ---- : --ROM  /b.v. de combikaart
      :  DATALIJNEN  --   :   RAM
      :    v.             :
      :  R/W              : --DISK-controller
      :  Ø2               :
      :                   : -- buffer --videobus --videocontroller
         inkomende  ---- :                  :        (ATOM-standaard)
         stuur            :                  :      --video-RAM
         signalen         :
                          : -- videocontrollers (club)
                          :                         (electuur)
                          :
                          :
                          : -- 6522 --------:      --printerbesturing
                          :                 I/Obus--overige I/O
                          :
                          : -- 8255 --------toetsenbord
                                            LS
                                            Cassette-i/o
```

1'STAP : PROCESSOR-KAARTJE.

Er is gekozen voor het plaatsen van de 6502,de buffers ,de klok, en de eerste deler op een apart kaartje (1/2 eurokaart).
Daarmee wordt het bovenstaande gerealiseerd met een minimum aan verandering van de huidige situatie,en waarbij iedereen naar eigen behoefte,kaarten kan gebruiken ,danwel tzt. kan vervangen.
(Zie schema 3 en printontwerp voor aansluiting aan PL7)
Welke wijzigingen op het moederbord zijn nu nodig ?
Zo moet IC4 vervangen worden door een "doorverbinder",die pen 1 en 19 ,2 en 18 ,3 en 17, etc,tot 8 en 12 doorverbindt.
Dan evt. IC5 eruit,evenals alle "bovenbouw" op IC 23 ,
en verder :
pen 8 van IC44 uitbuigen en de vrijgekomen plaats verbinden met punt 1-2 van de PL6 of evt PL7
Tot slot de 6502 uit de moederkaart lichten en op de processor-print plaatsten en de ombouw is klaar!

Aldus werd met verve gedaan,en gestart : helaas !;de bekende kreuL ACORN ATOM verschijnt niet.
De breaktoets heeft ook geen resultaat,en bij iedere opstart verschijnt een ander stramien van lijnen,letters,blokken,etc.

De vraag is nu,is er iemand,die,kijkende naar de documentatie, kan zeggen,waar een eventuele denkfout in dit verhaal zit.

In afwachting van respons,wordt alvast begonnen met de alternatieve versie van het processor-printje,nl. voor aansluiting op PL6, en dus binnen de kast.

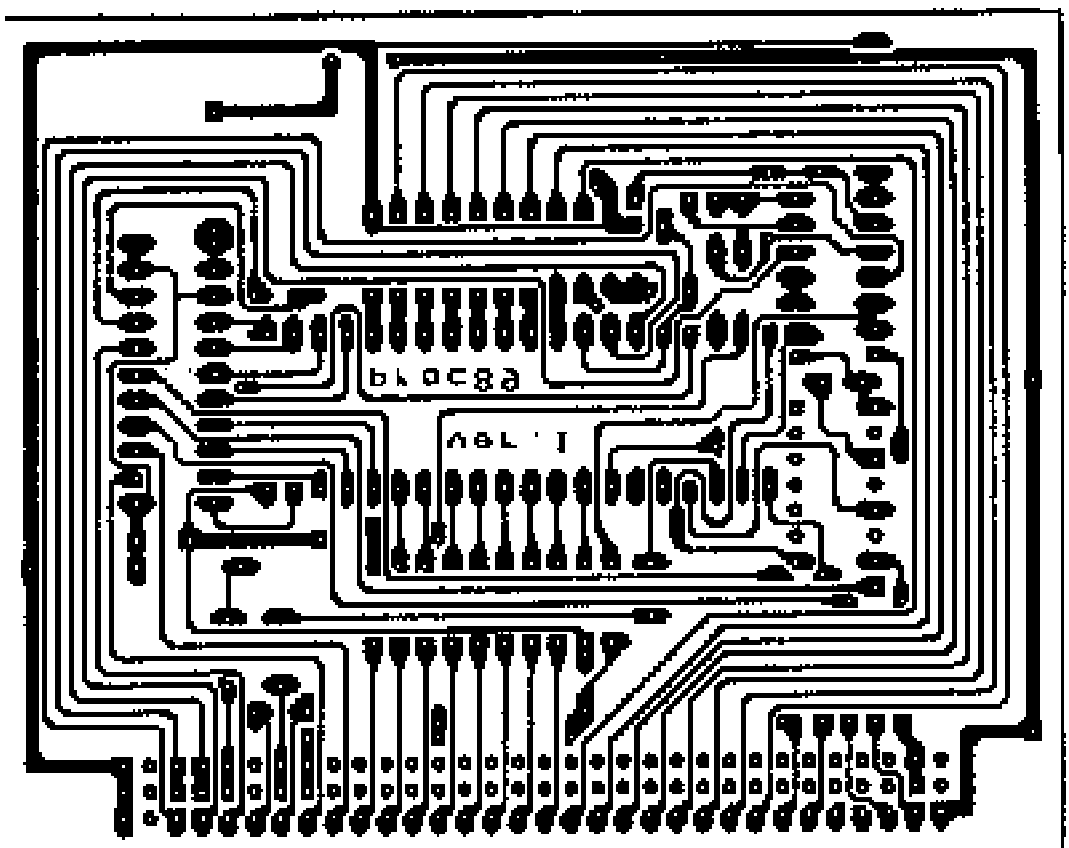
Z'STAP ; GEHEUGENKAART.

Verder is inmiddels al gereed ,maar nog niet gebouwd:

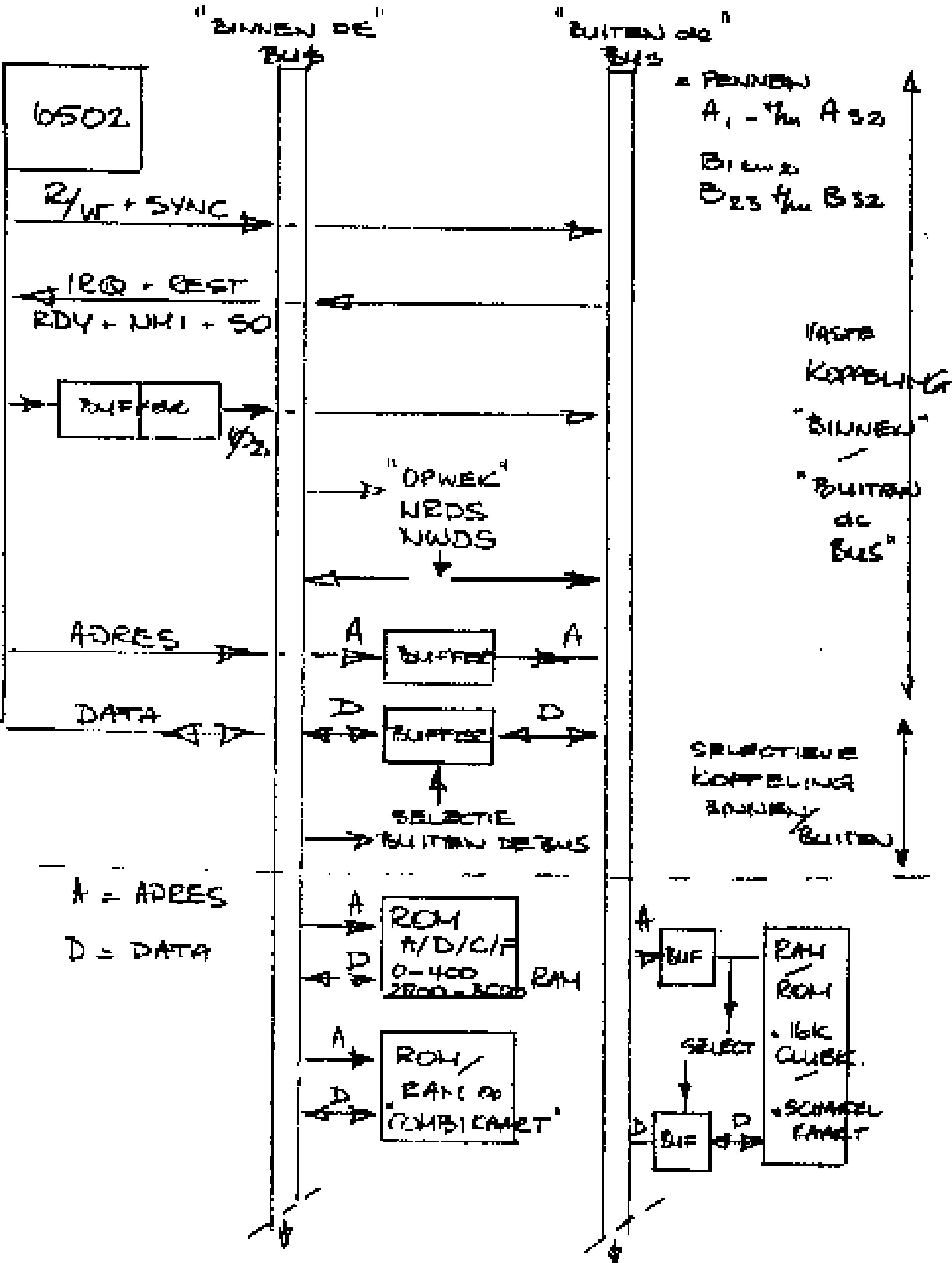
-de hedendaagse versie van de combikaart van Graus/Willems.
nl. opgebouwd op basis van 1* 32kCMOS (#0-#7FFF) 62256
1* CDEF in 27256 (evt. 2*)
8* A-ROM in 27256
2* A-RAM in 6264,
en dat alles direct op de ATOM-bus(PL7) aangesloten.
(enkelzijdig uitgevoerd!)

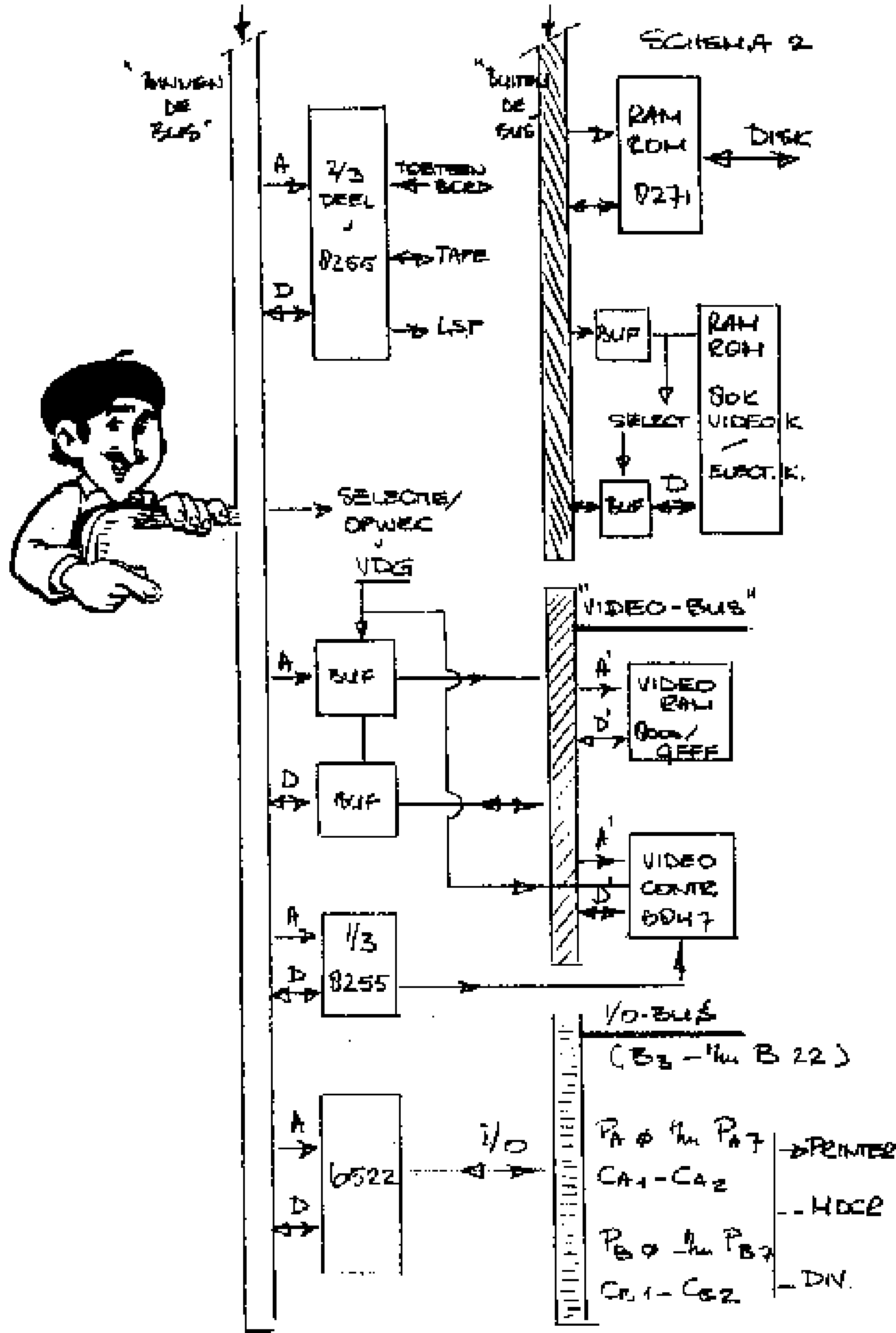
Tot slot wordt al gestudeerd aan de "rest-kaart";
nl. de 8255 met toetsenbord- en LS besturing,alsmede de orginele ATOM-videocontroller en -geheugen .

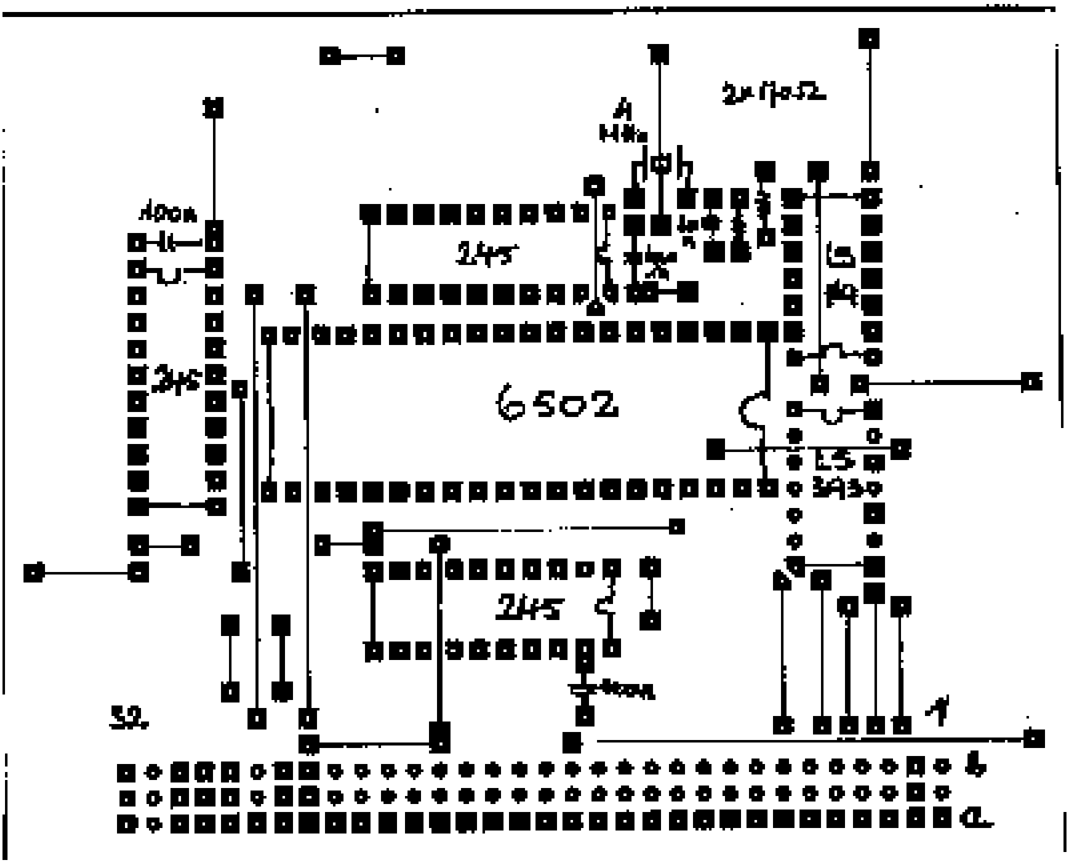
Bruno Tossaint



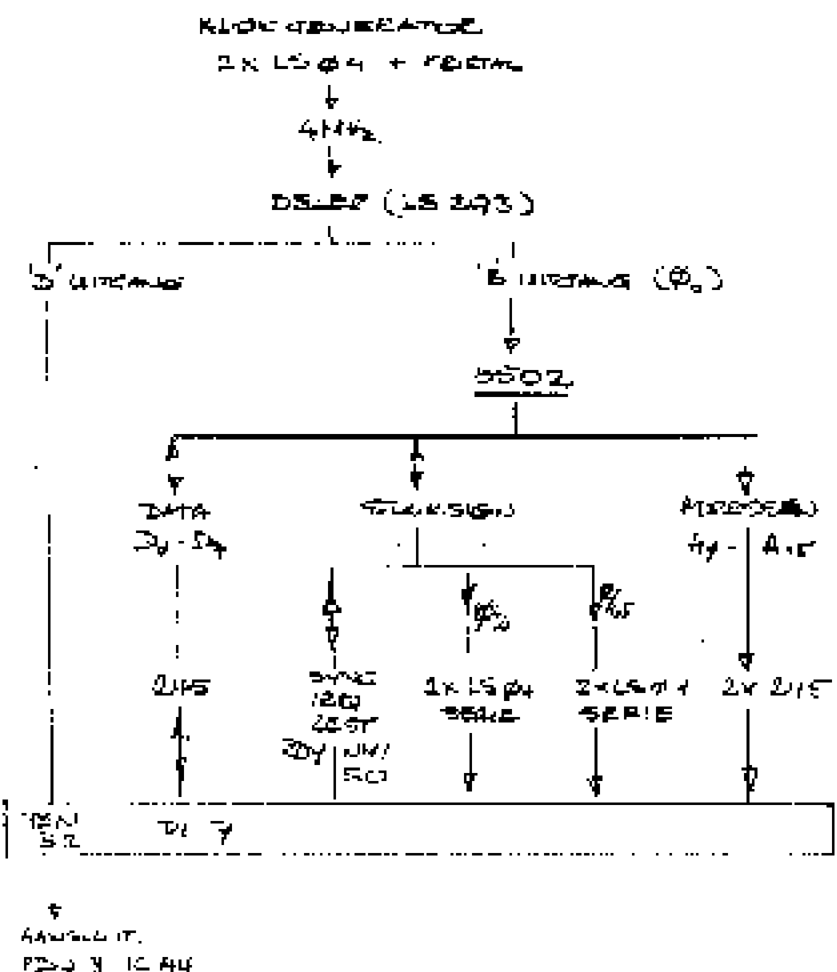
OVERZICHT ATOM. SCHEMA 1




SCHEMA 2
"BINNEN DE BUS"
"BUITEN DE BUS"
A
D
2/3 DEEL 8255
TOETSENBORD
TAPE
LSP
RAM ROM 8271
DISK
BUF
RAM ROM 80K VIDEO K / EUROK K.
SELECT
D
SELECTIE/OPWEK VDG
"VIDEO-BUS"
BUF
A'
D'
VIDEO RAM 8000/9FFF
VIDEO CONTR 6847
1/3 8255
I/O-BUS
(B3 -t/m B 22)
6522
I/O
PA ∅ t/m PA 7
CA 1 - CA 2
PB ∅ t/m PB 7
CB 1 - CB 2
PRINTER
HDCR
DIV.



1 PROCESSORKAART   SCHEMA 3



*** BALFAA 2.6 ***

INLEIDING

Al vrij snel na de introduktie van versie 2.5 ontdekte ik dat ASM-XTERN en .XTERN onder bepaalde omstandigheden niet goed werkten. Nu kun je moeilijk voor ieder kleinigheidje een nieuwe versie lanceren. Ik heb daarom in de loop der tijd een aantal zaken "opgespaard". Die zaken betreffen niet alleen "bugs" maar ook andere wijzigingen terwijl tevens een aantal nieuwe features zijn toegevoegd. Balfaa 2.6 is bedoeld als vervanger/opvolger van alle voorgaande versies. Er zijn louter normale 6502 instructies gebruikt zodat de assembler probleemloos op elke Atom te gebruiken is als toolkit op het A-blok.

BUGS

Zoals gezegd verliep de samenwerking tussen ASM-XTERN en .XTERN niet geheel vlekkeloos. Dit had o.a. te maken met de nieuwe mogelijkheid om het aantal externe tabellen aan te geven. Beiden zijn nu zodanig hergeprogrammeerd dat .XTERN met specificatie n ook werkelijk onder alle omstandigheden precies doet wat er verlangd wordt.

De oplettende Balfaa-gebruiker zal het misschien wel eens opgevallen zijn dat de symboltable na sorteren niet geheel volgens de regels gerangschikt is. De sorteer-routine UNRAVEL sorteert nu daadwerkelijk op alfabetische volgorde en is bovendien ongeveer een factor 2 sneller.

MELDINGEN

Bij het directive .HARDCOPY volgt geen melding meer als de printer om de een of andere redenen niet gereed is. In plaats daarvan is nu een time-out van circa 4 seconden ingebouwd.

Bij non-fatal foutmeldingen gaat de assembler nu zonder vragen verder. Bovendien wordt de OUT OF RANGE melding nu onderdrukt indien dit veroorzaakt wordt door een onbekend symbool. In dat geval wordt dus alleen de melding UNKNOWN SYMBOL gegeven.

De PAGE CROSS melding (waarschuwing) vindt nu alleen tijdens de 2e pass plaats. Bovendien wordt deze melding nu ook zonder listing weergegeven. Dit alles gebeurt uiteraard alleen als statusbit 5 gezet is.

De foutmelding ILLEGAL ADDRESSING MODE wordt nu tijdens de 1e pass onderdrukt indien deze veroorzaakt wordt door een te grote data-waarde in een instructie met 8-bits immediate adresseringsvorm. Dit komt van pas bij het laden van een register met een tabellengte min 1, bijv: LDX @end-begin-1. Als de tabel na de LDX instructie staat dan zijn de labels end en begin nog niet bekend. De assembler vult dan voor beiden #7FFF in met als gevolg dat de eindwaarde van de expressie -1 wordt ofwel #FFFF. Tijdens de 2e pass zullen beide labels bekend zijn en is er niets aan de hand. Hetzelfde zou een kunnen bereiken met LDX @<end-begin-1, maar dit heeft als nadeel dat de assembler ook tijdens de 2e pass geen foutmelding geeft als de tabel onverhoopt langer dan 256 bytes is.

NIEUWE DECLARATIE MOGELIJKHEDEN

Salfaa 2.6 kent de volgende declaratie mogelijkheden:

a. :label
b. :label = expr
c. :label(expr)
d. :label(expr1),expr2



a en b zijn oude bekenden. Bij a krijgt het label de waarde van de code-pointer P. Bij b krijgt het label de waarde van de expressie.

De varianten c en d zijn nieuw. Bij c krijgt het label de waarde van de code-pointer P, waarna de code-pointer met de waarde van de expressie verhoogd wordt. Indien de ram-pointer D actief is (dwz D<>#0000) dan wordt deze eveneens met dezelfde waarde verhoogd.

Variant d werkt hetzelfde als c maar nu worden de overgeslagen geheugen-plaatsen gevuld met de waarde van de 2e expressie. Als de ram-pointer actief is, worden de bytes vanaf D in het geheugen weggeschreven. Er worden uiteraard geen bytes gepooked als statusbit 1 op 1 staat.

Ter verduidelijking van de toepassingsmogelijkheden volgt nu een voorbeeld, links op de oude manier, rechts de nieuwe manier:

```
100 :werkgeh = #80                       100 .CODE #80
110 :buffer  = werkgeh+0                 110 :buffer(5)
120 :pointer = werkgeh+5                 120 :pointer(2)
130 :teller  = werkgeh+7                 130 :teller(1)
140 :vlag    = werkgeh+8                 140 :vlag(1)
150                                      150
160 .CODE #3000                          160 .CODE #3000
170 :start                               170 :start
180  LDA #0                              180  LDA #0
190  STA teller                          190  STA teller
200 :fill                                200 :fill(20),#EA \ 20 nop's
210 .BYT #EA;.IF P<fill+20 .GOTO 210     210  RTS
220  RTS
```

De nieuwe methode is compacter, duidelijker en handzamer. Stel dat later blijkt dat het buffer een byte te kort is. Rechts lossen we dat eenvoudig op met :buffer(6) maar links moeten we alle declaraties na het buffer met 1 gaan verhogen en dat kunnen er heel wat zijn. Bovendien is het om dezelfde reden gemakkelijker om rechts nieuwe labels tussen te voegen.

EXPLICIET EXTERNE LABELS

Met ::label kan men een label als expliciet extern declareren. Deze declaratie vorm is bij alle bovengenoemde mogelijkheden toegestaan.
Met ASM-XTERN zullen nu alleen de ::labels bewaard worden. Als er echter geen enkele ::label in de actuele tabel voorkomt, dan worden alle labels bewaard. Dit is gedaan om oudere sources runnable te houden.

Expliciet externe labels zijn in de eerste plaats bedoeld voor gebruik in lange sources welke uit meerdere autonome delen bestaan. Alleen die adressen welke we in latere delen nodig hebben geven we aan met ::labels. Ten eerste levert dit een aanzienlijke besparing op ten aanzien van de

benodigde geheugenruimte voor de symboltable. Ten tweede zal het assembleren van de latere delen behoorlijk wat sneller gaan. In het algemeen zullen we ::labels gebruiken voor belangrijke adressen zoals aanroepadressen van subroutines, begin-adressen van data-tabellen en adressen van werkgeheugenplaatsen. We krijgen zodoende automatisch een compacte en overzichtelijke tabel welke goede diensten kan bewijzen bij het testen en debuggen. Zo'n tabel is ook zeer geschikt om later als documentatie te gebruiken als een programma eenmaal voltooid is.

Natuurlijk kan men ook in een normale enkelvoudige source gebruik maken van ::labels met als doel om na het assembleren de mogelijkheid te hebben om onbelangrijke labels uit de tabel te kunnen verwijderen met ASM-XTERN.

Als we met .OPTION statusbit 4 op 1 zetten, dan zal de assembler alle labels die hij tegenkomt automatisch als expliciet extern opvatten, ongeacht of ze met : of :: beginnen. Dit is handig als we bij meervoudige sources naast een beknopte tabel ook nog een uitgebreide tabel willen hebben met daarin alle labels.

SYMBOLTABLE AFDRUKKEN

De manier waarop de symboltable door voorgaande versies van salfaa werd afgedrukt beviel mij gaandeweg steeds minder. Op een 40 koloms-scherm was het niet om aan te zien. Ook bij de printer moest men er altijd op letten dat de regelbreedte op een veelvoud van 16 ingesteld moest staan. Bij Salfaa 2.6 is het afdrukken van de tabel als volgt geregeld:

De systeemvariabele @ bepaalt het maximale aantal karakters dat per symboolnaam afgedrukt wordt. Te lange namen worden met een punt afgekort. De defaultwaarde 8 voldoet in het algemeen heel goed. Deze waarde is tevens als drempelwaarde gekozen zodat er altijd minimaal 8 karakters afgedrukt worden. Voor langere namen kan men @ naar wens in stellen.

Het aantal kolommen wordt bepaald door het laatste karakter van het betreffende statement of directive. Dit kan varieren van 1..9.
Als het laatste karakter voor de afsluitende ; of cr ongelijk is aan 1..9 dan geldt een default waarde van 2 kolommen. Enkele voorbeelden:

```
ASM-SYMBOLTABLE 5       ; 5 Kolommen
ASM-LAST 3              ; 3 kolommen
.SYMBOLTABLE            ; 2 kolommen (default)
.SYMBOLTABLE LAST 4     ; 4 kolommen
```

In direct mode is men gauw geneigd om opdrachten af te korten. Dit is geen probleem. Zo is ASM-S5 equivalent aan het eerste voorbeeld. Ik heb voor een default waarde van 2 kolommen gekozen omdat dit op elk soort beeldscherm een nette afdruk te zien geeft bij @=8.

BRK EN COP INSTRUCTIES

Dit zijn 2 gelijksoortige instructies. Beiden hebben nu 2 adresseringsvormen, te weten implied en immediate. Het is nu dus mogelijk om ook de BRK instructie van een signature-byte te voorzien, bijv: BRK #10.
Beide instructies kunnen dus naar keuze 1 of 2 bytes instructies zijn.

ASSEMBLER FUNCTIES

```
PASS       : geeft laagste 2 bits van het statusbyte.
WARNINGS   : geeft het aantal opgetreden waarschuwingen.
ERRORS     : geeft het aantal opgetreden fouten.
ATOP       : geeft het TOP-adres van de actuele symboltable.
XTOP(n)    : geeft het TOP-adres van de n-de externe symboltable.
```

In sommige sources wordt het zojuist geassembleerde programma direct opgestart met een LINK of CALL. Dit kan na eventuele non-fatal fouten voorkomen worden met bijvoorbeeld: IF ERRORS=0 THEN LINK start.

De functies ATOP en XTOP zijn hoofdzakelijk bedoeld om het automatisch saven en laden van tabellen op een meer flexibele manier mogelijk te maken. Daartoe wordt de aanwezigheid van OSCLI verondersteld.

BREAKVECTOR

Voor het opvragen van symbolen en functies is het noodzakelijk dat de breakvector naar de breakhandler van salfaa wijst. Salfaa kende daartoe reeds 4 statements welke de breakvector verzetten. Dit zijn SYM, CALL, ASM-BEGIN en ASM-CONTINUE. Er zijn nu 2 nieuwe mogelijkheden bijgevoegd:

Binnen een programma verzet het statement SALFAA de breakvector en doet verder niets. In direct mode drukt het commando SALFAA een overzicht af van alle statements en functies welke door salfaa herkend worden.

In direct mode heeft het gebruik van \ tot gevolg dat de breakvector verzet wordt. Het mag zonder afsluitende ; gevolgd worden door een opdracht. Bijv: \PRINT ERRORS (vergelijk p-stat bij P-Charme).
Binnen een programma kan \ gebruikt worden voor commentaar.

COMPATIBILITEIT

De P-Charme optie is niet meer aanwezig. De statusbits 6 en 2 hebben nu geen betekenis meer. Het hiermee samenhangende .M directive is eveneens vervallen. Dit waren min of meer ongedocumenteerde extra's welke waarschijnlijk alleen door enkele insiders gebruikt werden.

Labels mogen niet met functie namen beginnen. Wat dat betreft zouden de nieuwe assembler functies eventueel problemen op kunnen leveren. Gebruik in dat geval kleine letters of plaats ' voor het symbool.

Verder is salfaa 2.6 geheel opwaarts compatible t.o.v. versie 2.5. Voor het geval er zich onverhoopt toch problemen voordoen is het altijd handig als men kan controleren of de object code van salfaa intact is. De eenvoudigste manier is om alle 4096 bytes bij elkaar op te tellen. De laagste 2 bytes van de som moeten dan #407F zijn. Een meer betrouwbare test is de Cyclic Redundancy Check. Voor salfaa 2.6 is de CRC signature #36E9. Het is aan te bevelen om bij het in huis halen van salfaa een van de genoemde testen uit te voeren om er zeker van te zijn dat men in het bezit komt van een ongeschonden object code.

Literatuur:	SALFAA V0.4 V1.5	AN	jaargang 4	nr 2	blz 54..67
	SALFAA C2.0	AN	jaargang 5	nr 4	blz 17..22
	SALFAA 2.5	AN	jaargang 6	nr 2	blz 54..59

IN DE HOOFDROL :

```
++++++++++++++++++++++++++++
+   RANDOM  ACCESS  FILES  +
++++++++++++++++++++++++++++
```

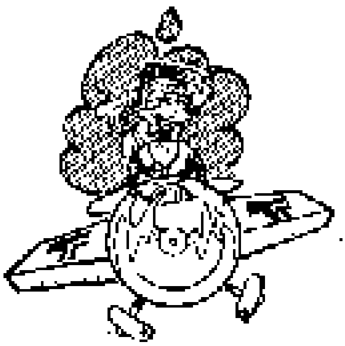
presentatie : Roland Leurs

In dit artikel wordt niets uitgelegd over het gebruik van random files, maar worden enkele korte, misschien ook nog nuttige, aangeboden. Voor de theorie van random access files verwijs ik u naar AN3-4 pag.31.

FILE KILLER

Sommigen onder ons bewaren nog steeds belangrijke en/of persoonlijke gegevens op schijf. Als deze data verwijderd moeten worden kan de file gewist worden met het DELETE commando. Maar in werkelijkheid wordt niet de file verwijderd, maar de filenaam. Maar dat wist u al.
Dit programma schrijf m.b.v. een random access file echt alle data kapot, zodat niemand ooit nog iets terug vindt op de schijf. En u ook niet, dus denk goed na voordat u het programma gebruikt.

```
 10 REM FILE KILLER
 20 DIM F(7)
 30 INPUT "WELKE FILE MOET WORDEN VERWIJDERD "$F
 40 A=FIN $F
 50 IF A=0 THEN PRINT "FILE BESTAAT NIET !"';END
 60 L=EXT A
 70 FOR X=0 TO L
 80 BPUT A,#2A
 90 NEXT X
100 *SHUT 0
110 $#100="DELETE "
120 $#100+LEN #100=$F
130 LINK #FFF7
140 END
```



EDIT-SP

Het tweede programma is bedoeld als hulpje voor de editor ED64/ED80. Zoals in de handleiding van Atomdos niet staat geschreven kan met het commando *SPOOL een tekstfile gemaakt worden. Als we deze tekstfile met de editor willen bewerken gaat wel alles fout. Dit is het gevolg van de aanwezigheid van diverse control codes.
Onderstaand programma leest alle karakters uit de opgegeven tekstfile en copiert die naar een tweede file. Echter alle verboden codes worden vervangen door spaties. Hiervan merkt u niets in de tekst die u nu wel met de editor kunt gebruiken.
De oorspronkelijke tekstfile blijft ongewijzigd op de schijf achter.

```
 10 REM SPOOL-FILE AANPASSING
 20 DIM F(7),N(7)
 30 INPUT "WELKE FILE MOET AANGEPAST WORDEN "$F
 40 INPUT "NIEUWE FILENAAM "$N
 50 A=FIN $F;L=EXT A
 60 IF A=0 THEN PRINT "FILE BESTAAT NIET !"';GOTO 150
 70 B=FOUT $N
 80 FOR X=0 TO L
 90  Y=BGET A
100  IF Y=13 THEN GOTO 120
110  IF Y<32 OR Y>126 THEN Y=32
120  BPUT B,Y
130 NEXT X
140 BPUT B,4
150 *SHUT 0
160 END
```

VECTORIZER

Tot slot een handig programma dat te vergelijken is met een primitieve sourcemaker. Een voorbeeld is hier misschien de duidelijkste manier om de werking uit te leggen.
Als we dit programma loslaten op een stuk geheugen, dan wordt de file 'VECTOR' op schijf aangemaakt. Van deze file kunnen we met de opdracht '*EXEC VECTOR' de file omzetten in een Basic programma. Dit basicprogramma heeft de volgende vorm:

```
    10 X=#8000
    20 X!0=#38DB6A9B
    30 X!4=#FF05FF44
       ... etc
```

Een soortgelijk programma staat ook in het Electuurboek 'Programma's voor de Acorn Atom'.
Het spreekt voor zich dat er voor het uiteindelijke programma genoeg geheugen aanwezig moet zijn. Er kan ongeveer 4 K geheugen omgezet worden tot een programma met vectoren als in de Atom geheugen is van #2900 t/m #7FFF.
Op de regioschijf staat een voorbeeld programma van deze vectorizer. Het betreft een clear4 tekening van 6 Kbytes lang. De "source" hiervan is echter bijna 30 Kbytes lang en is daarom opgesplitst in twee files. Om de tekening terug te halen typt u alleen in *RUN SURF1 en het programma doet de rest.

```
 10 REM VECTORIZER   V1
 20 REM ROLAND LEURS - NATTENHOVEN
 30 INPUT "BEGINADRES "B
 40 INPUT "EINDADRES  "E
 50 INPUT "EERSTE REGELNR"R
 60 INPUT "STAPGROOTTE   "S
 70 *SPOOL *
 80 @=0;T=0
 90 PRINT R" X=#"&B';R=R+S
100 FOR X=B TO E STEP 4
110 PRINT R" X!"T"=#",&!X'
120 T=T+4;R=R+S
130 NEXT X
140 PRINT R" END"'
150 *SHUT 0
160 PRINT ''"EVEN GEDULD A.U.B."' ;@=8
170 A=FIN "*"
180 L=EXT A
190 B=FOUT "VECTOR"
200 FOR X=0 TO L
210 PRINT $13,L-X
220 Y=BGET A
230 IF Y=13 THEN GOTO 250
240 IF Y<32 OR Y>126 THEN Y=32
250 BPUT B,Y
260 NEXT X
270 *SHUT 0
280 *DELETE *
290 PRINT ''"MAAK NU HET PROGRAMMA MET"'
300 PRINT "   *EXEC VECTOR"'
310 END
```

Tot zover deze programma's met random access files. De programma's op de regioschijf die bij deze tekst horen zijn:

FKILLER	programma zoals in tekst is beschreven
EDIT-SP	programma zoals in de tekst is beschreven
VECTORI	programma zoals in de tekst is beschreven
SURF1	demonstratie vectorizer deel 1 (*RUN)
SURF2	demonstratie vectorizer deel 2

Nieuws over RSCOM

Toen het artikel over RSCOM (A.N. 89-1) gepubliceerd was, was het eigenlijk al weer verouderd. De meegeleverde versie was toen versie 2, die dateerde van december 88. Toen Atomnieuws in de bus rolde was het april 89, en de ontwikkelingen hadden, en hebben niet stilgestaan. In april werkte ik zelf al met versie 5, en die is nu ook al weer overboord. Versie 6 is het nu, en dat zou het wel eens kunnen blijven. Op een goede X-modem up- en download-routine na zit alles wat ik wilde er wel ongeveer in. Aan dat laatste wordt nu dus gewerkt, hetgeen uiteindelijk moet resulteren in versie 6X: versie 6 met X modem software.

Veel van de wijzigingen zijn aangebracht nadat in de praktijk problemen met een bepaald programmadeel werden gekonstateerd. Andere resulteerden uit mijn eigen ontevredenheid over de werking en de struktuur van het geheel. Ook heb ik naar aanleiding van op en aanmerkingen van anderen nog enkele wijzigingen aangebracht. Zodoende krijg je een soort evolutie in je programma, waarbij je jezelf soms afvraagt: waarom heb ik dat niet eerder zo gedaan? Voor mij staat het vast dat het oorspronkelijke programma sterk verbeterd is, zoals de opsomming hieronder aangeeft:

Veranderingen en wijzigingen t.o.v. Versie 2

1: Alle control-codes kunnen nu vanaf het toetsenbord verzonden worden. Terminalfunkties worden nu via de REPT-toets verzorgd.

2: De kommando's zijn gewijzigd. Door een andere manier van instellen te gebruiken, zijn veel kommando's vervallen, en de overgeblevene zijn korter geworden.

3: Indien als VDU-driver het programma VIDIVDU van Dick Bronsdijk gebruikt wordt, is het mogelijk om op de 80-kolomskaart viditelbeelden weer te geven.

4: Het programma is vanuit machinetaal eenvoudiger aan te roepen.

5: Het programma is korter geworden, en het geheugengebruik in het gebied 21C-... is iets minder.

6: Als extra is het programma uitgerust met X on X-off software handshaking. Hierdoor is het mogelijk om de printer mee te laten lopen, ook als het een trage is.

VERKLARING

Punt 1: Om de extra funkties van de terminal te besturen werd oorspronkelijk gebruik gemaakt van de CTRL-toets. Dit had

als nadeel dat controlcodes (CTRL+ toets) niet gegenereerd konden worden. In de nieuwe versie scant het programma de REPT toets igv de CTRL. Deze komt zodoende dus vrij.
Stoppen van het terminalprogramma gaat nu dus met REPT + Q (was CTRL + Q).

Punt 2: Het aantal gebruikte kommando's was te groot, en enkele zaken konden onder bepaalde omstandigheden wel eens fout gaan. Deze fouten herstellen zou veel extra geheugenruimte vergen, en het geheel was naar mijn smaak al wat aan de lange kant. Als oplossing voor deze problemen heb ik een nieuwe kommandostruktuur bedacht. Alle instellingen in het programma worden nu verzorgd via menu's. Dit zijn blokken van acht bytes, die alle instelbare parameters bevatten. In deze acht bytes zit dus de zend en ontvangstsnelheid, de pariteit, het aantal databits enz. RX, TX, FMT en RST. Nog wel in gebruik zijn: TER, TCPY en NCPY. Bij TCPY en NCPY is de Y vervallen, drie letters leek mij duidelijk genoeg. Als nieuw kommando is toegevoegd: MENU.
Dit kommando heeft de syntax: MENU (waarde van 0-31)
De andere kommando's hebben geen parameters meer, na het kiezen van een bepaald setup-menu ligt alles vast.

Hoe werkt dit alles?
Met behulp van het kommando MENU wordt een keuze gemaakt uit een aantal vooraf gekozen instellingen. Wanneer nu TER of TCP aangeroepen wordt, dan wordt de gekozen instelling in de gekozen ACIA-kaart geplaatst, en het spul gaat aan de slag. Opmerking hierbij is dat de gekozen ACIA kaart ook al vastligt in het gekozen MENU. Redenering hierachter is, dat op deze wijze voorkomen wordt dat de verkeerde poort wordt aangeroepen. Je kunt nu eenmaal niet 1200/1200 werken over een 300/300 modem.
Ook ligt het type VDU vast in de setup. Bij het initialiseren wordt dan automatisch ook de juiste VDU gekozen. Mogelijkheden zijn 40 of 80 koloms, of viditelscherm.
In het laatste stuk van dit verhaal staat beschreven wat er met de setup-menu's allemaal is vast te leggen, en hoe dit dient te geschieden.

Punt 3: In het vorige artikel stond dat er gewerkt werd aan een mogelijkheid om viditel te draaien. Dit is nu mogelijk, door het programma VIDIVDU van Dick Bronsdijk als videoaansturing te gebruiken. SHIFT-O geeft het viditelhekje (code #5F).

Punt 4: Vanuit machinetaal is het programma op de volgende manier toegankelijk: Het statement MENU zet op lokatie #21C het nummer van het gekozen menu (0-31) weg. Dit is de enige funktie van het statement MENU. Vanuit een machinetaalprogramma kan dus volstaan worden met het invullen van #21C, gevolgd door een sprong naar het startadres van de terminal.

Punt 5: Door het laten vervallen van kommando's is veel ruimte vrijgekomen, niet alleen de kommando's zelf verdwijnen, maar ook de foutkontrole's. In totaal is het programma ongeveer 300 bytes korter geworden op een oorspronkelijke lengte van ongeveer 1200 bytes, terwijl ook nog software handshaking is ingebouwd. Dit mag dus aanzienlijk genoemd worden.

Punt 6: Software-handshaking wordt toegepast bij datacommunicatie via bijvoorbeeld een telefoonlijn. Als de ont-

vanger het niet maar bij kan houden stuurt hij een X-off karakter naar de zender. Als het verzenden weer hervat kan worden, stuurt de ontvanger weer een X-on naar de zender. RSCOM stuurt een X-off karakter als de buffer 128 karakters of meer bevat. Dit is 50% van het maximum aantal, dus ruimte genoeg, ook bij hogere snelheden. Als de buffer helemaal leeg is wordt weer een X-on verstuurd. In het statusbyte van de terminal is dit te volgen via twee bits die gezet worden als een X-on resp. X-off verstuurd worden. Ook is deze funktie via een bit in datzelfde statusbyte aan of uit te zetten.

Tot zover de grote wijzigingen in het programma. Er zijn nog een aantal kleinere, waarvan ik er hier een paar even noem:

-Alle funkties van de terminal zijn via een enkel statusbyte te kontroleren.

Om de linefeed wel of niet weer te geven, wordt OSWRCH op twee manieren aangeroepen (#FFF4 of #FFED). Welke van de twee hangt af van het ingesteld zijn van wel of geen linefeeds weergeven. Als geen linefeeds weergeven ingesteld is, dan volgt alleen na elke RETURN automatisch een linefeed, en verder nooit.

-De DELETE-toets genereert nu een backspace (CTRL-H). Dit wordt door alle mij bekende databanken geaccepteerd. Zodoende is het mogelijk om tikfouten te herstellen.

-Ten overvloede nogmaals: SHIFT-0 geeft nu het viditelbekje weer.

-Om ook automatisch verzenden te kunnen gebruiken bij BBSen met een lage verwerkingssnelheid (Commodore 64 en consorten), is een vertraging aan het eind van een regel ingebouwd. Deze is in te stellen tussen 1/60 en 255/60 sekonde. Deze routine maakt gebruik van de bekende vertragingsroutine in de ATOM op #FB83.

-Alle aanwezige kaarten in het systeem gebruiken nu #600 als ontvangstbuffer. De pagina #700-#7FF is hierdoor momenteel weer vrij. Het ligt in mijn bedoeling om te zijner tijd deze pagina als TRANSMIT buffer te gaan gebruiken.

Tot slot nog de beschrijving van de instellingen in de setup: Deze setup bestaat zoals gezegd uit 8 bytes. Deze bytes hebben de volgende funktie:

BYTE	MIN/MAX	Beschrijving
0	#00-#80	Offset, duidt een bepaalde kaart aan. Kaart 1 zit op #B900, kaart 2 op #B920 Offset loopt dus op in stappen van #20
1		Specificeert de inhoud van dataregister A van de VIA: b0,b1,b2: Snelheid externe klok b3 Externe klok aan of uit b4 DCD constant aan b5 DSR constant aan

2	#01-#FF	VIA timer1 High latch. Hierin is de gewenste vertragingstijd voor het opnieuw lezen van het statusregister van de ACIA in te stellen.
3		ACIA command register. Zie datasheet 6551
4		ACIA control register. Zie datasheet 6551
5		Terminalstatusbyte: b0: X on verzonden b1: X-off verzonden b2: vrij (gereserveerd voor X-modem) b3: autotransmit aan/uit b4: logfunktie aan/uit b5: local echo aan/uit b6: linefeed enable/disable b7: X-on/X-off aan/uit
6	#01-#FF	Timeout1: Vertraging aan einde regel
7	40,60,80	VDU type: 40 of 80 koloms VDU. Keuze 60 is viditelscherm (mits aanwezig)

Tot besluit nog enkele tips om zelf menu's aan te maken. Ten eerste altijd zeer zorgvuldig tewerk gaan. Het aanmaken van deze menu's is een precies werkje, het hele funktioneren van het programma is ervan afhankelijk. Het beste is het om de gewenste instellingen eerst op te schrijven in de juiste volgorde, en daarna byte voor byte te koderen. Bij problemen de datasheet van de ACIA raadplegen. Voor de meeste databanken KAN de linefeed aan staan, voor VIDITEL-protocol MOET hij aanstaan. In het algemeen kun je stellen dat de linefeed aan dient te staan, als de te raadplegen databank hem verzendt. Een reden om hem uit te laten is om lege stukken in de tekst te onderdrukken. Ik gebruik dit bijvoorbeeld voor MEMOCOM. Ook bij "wazige" BBS'en kan overwogen worden om de linefeed uit te laten. Je bent er dan zeker van dat er geen regels overschreven worden. Verder moet er goed worden opgelet dat voor VIDITEL de goede VDU driver (VIDIVDU) ook werkelijk in het geheugen zit. Als deze niet op het defaultadres (#7000) geladen wordt, moet in de source van RSCOM6 het label VIDVID aangepast worden. Het defaultadres voor VIDIVDU is slechts tijdelijk. Als alles helemaal klaar is, moet VIDIVDU met RSCOM in een ROM komen te zitten. Waar ik die dan ga laten, dat zien we dan wel weer.

Tot zover het vervolg op RSCOM. Zelf had ik niet verwacht dat dit vervolg al zo snel zou moeten verschijnen, maar ja, de ontwikkelingen in ATOM-land gaan vaak sneller dan men denkt.
Voor vragen en kommentaar ben ik nog steeds te bereiken op onderstaand adres: Regio Zeeland

UITBREIDINGEN OP COMMUNICATIEPROGRAMMA RSCOM6.

Het communicatieprogramma RSCOM6 is een knap stukje werk van Leo Gijssel, ook Regio Zeeland, dat zeer goed werkt.
Ik heb echter een hekel aan het inloggen bij databanken, dit moet automatisch gaan. Een hulpprogramma hiervoor heb ik al geschreven voor VDT6. van de Otten fabriek en dit moest eenvoudig aan te passen zijn voor RSCOM6.
Leo liep echter met het idee rond om ook het draaien van het telefoonno. automatisch te laten doen door zijn DIAL-1.4 programma.
Na rijp beraad hebben we toen besloten dat ik beide hulpprogramma's zou combineren tot een uitbreiding voor RSCOM6.
Het volgende is eruit gekomen en heeft de nodige telefoontjes gekost.

RSCOM6 heb ik aangepast v.w.b. het beginadres van de LOG-optie en tevens dat van de AUTOTX optie nl. op #2A00.
Het programma is omgedoopt tot V24-COM. en bevat op dit moment de voor mijn systeem geldende adressen (**!!**).
RSCOM6 heeft functietoetsen d.m.v. <REPT><letter>, V24-COM. heeft deze ook, maar is uitgebreid met <REPT><D> voor autodial&login.

Bij deze functietoets kom je in het menu met de databanken.
Na het intikken van het nummer voor de betreffende databank wordt deze gebeld, zet nu het modem evt. on-line. Na het draaien van het no., met herhalings-mogelijkheid bij 'in-gesprek', zal de ACIA gezet worden indien de telefoon 'overgaat' of er een modem aan de andere kant opneemt.
(Als er echter storing op de lijn zit kun je dit ook bereiken door <CTRL> in te drukken.)
De ACIA zal nu wachten op een DCD (carrier detect) gedurende max. 10s.
Is deze er niet dan zal de melding 'no carrier detect' komen en kom je terug in het databanken menu.
Is er wel een DCD dan zal de terminal ingesteld worden voor de betreffende databank.

De terminal werkt en de databank meld zich. Toets nu <REPT><A> en de inlogcodes worden verzonden.
Gemakkelijker kan het niet. praktijk is simpeler dan de theorie.
Na het einde van de verbinding <REPT><D> en je komt weer in het databankmenu.

De 'in gesprek' detectie en afhandeling zorgen ervoor dat de centrale denkt dat de verbinding verbroken is, terwijl ondertussen het modem nog 'online'
staat. Indien je nu na 20s. wachten het programma opnieuw laat bellen zal er weer een kiestoon zijn.
Indien je besluit om met de <REPT> toets de herhaling te stoppen kom je weer in het databank menu, laat de modem gerust 'online' en kies een
andere databank.

Op het allerlaatste moment heb ik nog een aardigheidje ingebouwd indien je Big-Benny bezit. Het programma gaat er vanuit dat je de BB uitleesroutine al ergens hebt staan, bij mij op #0F00. vul dit in bij de D&L-SRC.
Het 'modemen' begint nu weer vanuit BOOT-V24 welke nu de adressen #9000 t/m #9050 op #00 zet. Na het draaien van een nummer en connect' zal de tijd opgeslagen worden. Indien de sessie met de databank is beeindigd gewoon <REPT><D> om in het menu te komen en je ziet de TOTAAL verbelde tijd na het gebruik van BOOT-V24. Indien je langer dan 12 uur dezelfde databank belt, bestaat de kans dat de 'tijd' die wordt aangegeven minder is dan de gebelde. De regels zijn in het programma gemerkt met *!BB!* en aflijning.zodat je ze eventueel kunt verwijderen bij gebrek aan Big Benny.

De programma's die dit verzorgen zijn D&L-FILES, D&L-SRC, D&L-ASS en BOOT-V24.
D&L-SRC is de source. hierin vul je de namen van de jou bekende databanken in en je 'runt' het. Dan geef je *S. D&L-ASS 9100 995E (kijk zelf nog even voor het eindadres) en de helft is al gebeurt.
Met D&L-FILES maak je de kleine databank-bestandjes aan met telefoonno., inlogcode en terminal instelling.
De baudrates en instellingen die ik gebruik zie je in D&L FILES wanneer er om gevraagd wordt.
Bevalt dat je niet dan moet je V24-COM. aanvullen met installingen, max. 10 mogelijkheden zonder herschrijving van D&L-SRC.

BOOT-V24 tenslotte start het hele spul voor de eerste maal op na het inschakelen van je ATOM. geef *BOOT V24 en het werkt.
Nu wordt D&L ASS en VIDIb (videotex emulatie voor 80 koloms, ook Leo) geladen, er verschijnen tevens enkel teksten o.a. over de functietoetsen.
Na enkele toetsaanslagen kom je in het databanken menu en kun je beginnen.

Voor info over V24 COM, zie de uitleg van Leo Gijssel v.w.b. RSCOM6.

Voor vragen cq. verbeteringen bel gerust:

Roger Boussen. Leo Gijssel.

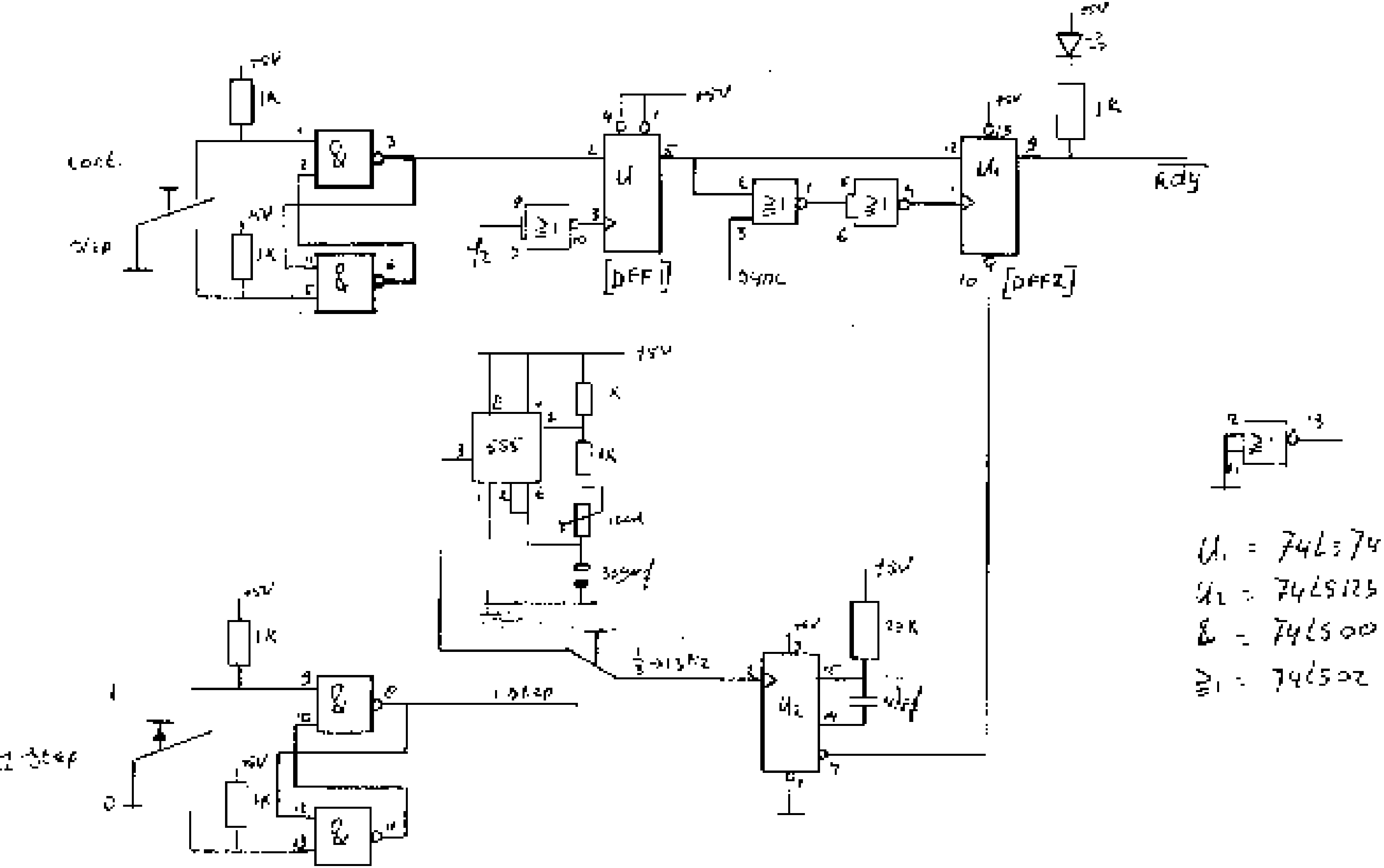
Voor de harde jongens

HARDWARE 6502- tracer

J. Swinkels

Tijdens het testen van mijn "Euro Atom" had ik de behoefte aan een "step mode",waarmee je een programma stap voor stap kunt laten runnen.Iedereen denkt dan natuurlijk meteen een de josbox, waar zo'n statement inzit. Helaas werkt mijn gehele systeem niet,waardoor dit dus uitgesloten wordt.
Ik ben toen eens in verschillende electuurs gaan zoeken en vond daar allerlei "6502 tracers",zoals ze die noemen, maar deze werkten allen op interrupt basis,iets waar ik dus geen fluit mee kon doen.
De conclusie die ik hieruit trok was dat er dus geen echte hardware tracer voor de 6502 bestond. Dus wat doet een hardware man dan, die ontwerpt, maakt en gebruikt zijn eigen ontworpen tracer.
Zo gezegd, zo gedaan en zie hier. Naar alle waarschijnlijkheid de eerste hardware 6502 tracer in Europa!.
Aan alleen een tracer heb je niet heel erg veel, omdat je dan telkens aan de adres- en data-bus moet meten om er achter te komen op welk adres welke data staat. Daarom heb ik meteen ook maar een status-analyser ontworpen die het adres op de bus omzet in een leesbaar adres op het display.
Hetzelfde heb ik met de databus gedaan. Door nu de cpu step by step te laten lopen zie je telkens welk adres en data op de bus staat.
Dit is niet het ei van columbus maar de 6502 cpu tracer van Jan Swinkels .
De kosten van deze tracer zijn ongeveer FL.115,00 incl. print.
Snelle hardware debugging van een programma is nu mogelijk zonder dat je gebruik van je interne registers maakt, waardoor het progamma onverstoord verder werkt.

Zijn er personen die menen dat een 6502 processor ook te besturen is met een One-shot op de klokingang (pen 37), dan hebben ze een grote uitvinding gedaan want volgens de data sheets kan dit niet.
Dit heb ik al reeds geprobeert en inderdaad bevestigd.
Bij een klokfrequentie lager dan 843 KHz, is een 6502 processor niet meer thuis.
Bij deze tracer wordt, d.m.v. een signaal dat er speciaal voor ontworpen is, de cpu stilgezet, terwijl de klok nog doorfietst op 1-of 2-MHz.
Dit Ready-signaal is bij een Z-80 processor ook aanwezig maar heet dan Halt. Het is dan rechtstreeks met een OR-poort aan te sturen zonder dat er eerst gekeken wordt naar andere signalen. Dit wordt intern allemaal geregeld.
Een 6502 is in principe normaal een wat oudere processor, maar daardoor nog lang niet de slechtste.



Jan Swinkels
Dorpsstraat 41
6093 EA Heythuysen
TEL. 04749 1235

Werking Tracer en Status Analyser.

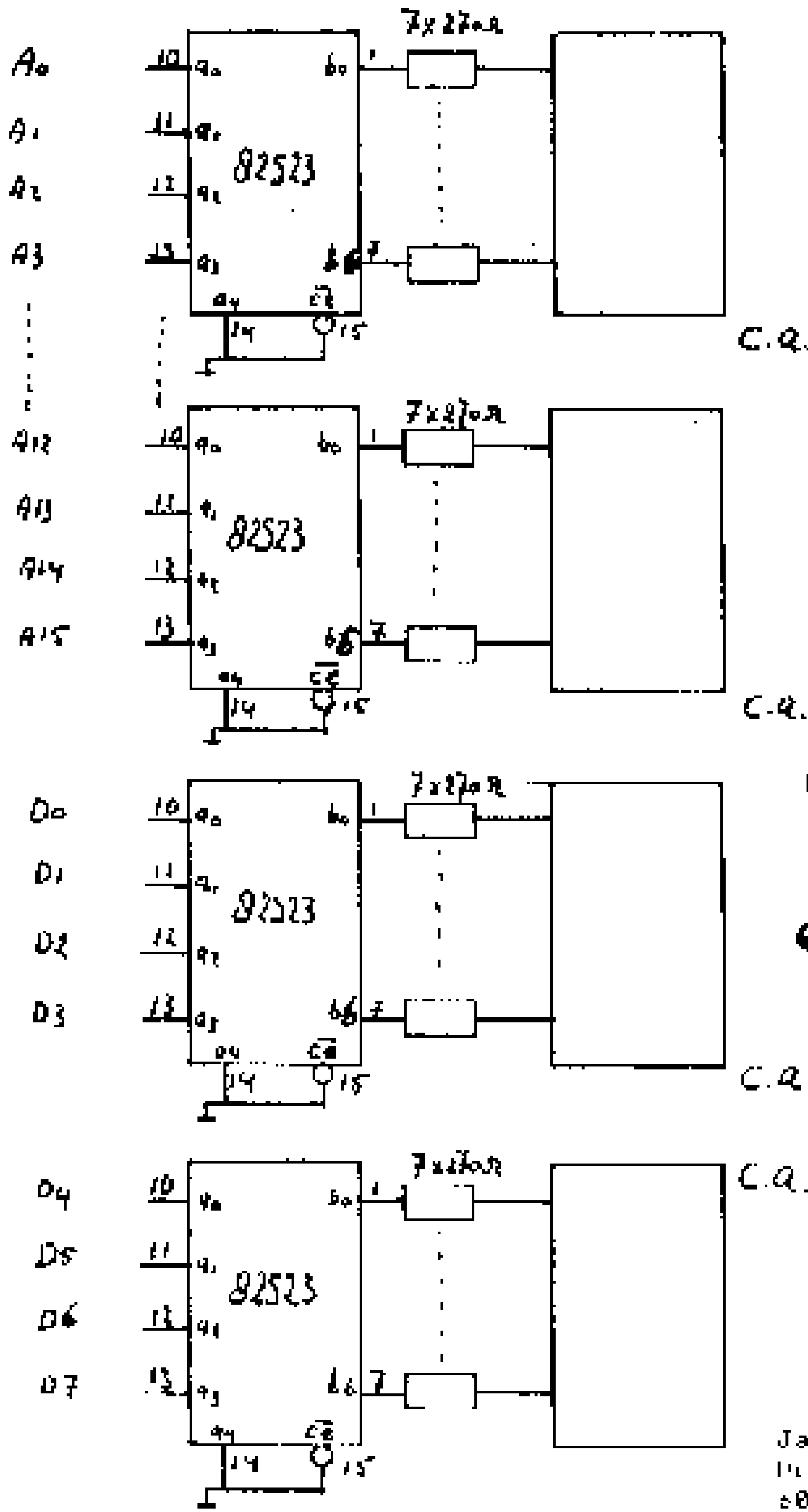
Op de tekening zijn twee schakelaars te zien. S voor in de step-mode en C voor de cpu een stap verder te verplaatsen.
Indien de spanning op het systeem gezet wordt, dan zal de ready lijn hoog worden (er wordt automatisch een 1 in de data flip-flops [DFF] 1 en 2 geklokt). Hierdoor zal het systeem normaal opstarten.
Indien nu de schakelaar S op de step-mode gezet wordt dan zal uitgang Q een 0 worden. Deze 0 wordt aan de ingang van DFF1 aangeboden. Indien nu een instructie wordt opgehaald, zal fi 2 laag worden. De geinverteerde fi 2 zal dus hoog worden en deze 0 inklokken.
Deze nul komt nu aan de ingang van DFF2 te staan. Doordat de uitgang van DFF1 nu laag wordt zal de Nor-poort niet meer geblokkeerd zijn voor het sync signaal.
Enige tijd nadat fi 2 laag geworden is (350 nsec max) zal de sync hoog worden.Hierdoor zal de data in DFF2 geklokt worden. Doordat deze data 0 was, zal de Ready lijn laag worden en dus de cpu stoppen. De cpu stopt dus op het moment dat de cpu een fetch heeft gedaan.
De cpu staat nu volkomen stil en op het display is nu de status van de cpu af te lezen. In deze stand kan de cpu gerust een hele nacht verblijven (heb ik uitgetest!).
Door nu op de schakelaar C te drukken zal de MMV een set-signaal genereren van ongeveer 40 nsec waardoor er een 1 in DFF2 wordt geklokt, de cpu loopt weer verder (schak. C is een terugverend type).
De instructie waarmee de cpu juist begonnen is wordt nu afgemaakt. Als deze instructie afgelopen is, wil de cpu weer een volgende instructie ophalen, echter het Sync-signaal klokt weer een nul in het DDF2 waarmee de cpu zichzelf weer stilzet.
Op deze wijze is het dus mogelijk om step-by-step door een programma te lopen.

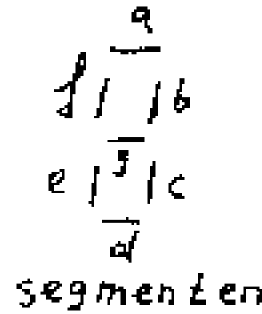
Bij een schrijfactie zal het Ready-signaal automatisch intern in de cpu genegeerd worden.
Als we de cpu weer "op volle toeren" willen laten draaien dan schakelen we schakelaar S weer terug in de stand Continue en er gebeurt niets. Dit klopt want op dit moment staat de cpu nog stil.
Wordt op de schakelaar C gedrukt, dan wordt DFF2 geset en daardoor de cpu weer opgestart. Bij de eerst volgend instructie zal deze 1 in DFF1 ingeklokt worden. Het Sync-signaal wordt weer hoog en automatisch deze 1 inklokken in DFF2.
Hiermee wordt automatisch de Nor-poort gesloten waardoor de cpu nu niet meer in de step mode staat.

Schakelaar "var." dient ervoor om in de step-mode, de cpu weer automatisch op te laten starten. D.m.v. de potmeter is de frequentie in te stellen van 0.3 Hz to 13 Hz. Je hebt dan een volautomatische stepper Bij lange programma's is dit gemakkelijk en kost je geen blauwe vinger meer.

De tracer werkt uitstekend. Alleen is er nu nog het probleem dat ik steeds met een scoop op de processor moet gaan meten om erachter te komen op welk adres , welke data staat.
Omdat dit een zeer tijdrovend en vervelend werkje is ben ik op zoek gegaan naar IC's die een hexadecimale code omzetten in een zevensegments code. IC's hiervoor bestaan er wel maar, omdat die niet vaak gebruikt worden, zijn deze peperduur FL 11,78 per stuk (ik heb er 6 nodig)."$$$$".
Ik ben toen eens gaan informeren wat prom's kosten. Deze bleken de helft goedkoper te zijn en heb daarom besloten om zes prom's te kopen en ze dan zelf te programmeren.
Van een oude wekker heb ik het display afgesloopt en met nog wat weertandjes heb ik nu een complete status-analyser gemaakt.
Met de tracer zet ik de cpu stop en de analyser zet de gegevens van de cpu om op een zevensegments display.
D.m.v. een schakelaar is het mogelijk om de cpu continue te laten werken of in de step-mode. Met een klein drukknopje is het nu mogelijk om in de step-mode de cpu step-by-step te laten lopen. indien de cpu op "volle toeren" draait, dan zulen alle display's het cijfer 8 laten oplichten.
In de step-mode kan precies worden nagegaan op welk adres de cpu stilstaat en op het data display is dan de data van de cpu zichtbaar. Het mooie van deze analyser is dat bij een read- of write-operatie de data niet door de analyser beinvloedt wordt,iets waar ik eerst bang voor was.
Nu is het ook voor een hardware man mogelijk om eens te achterhalen hoe het operating-system werkt, zonder dat hij zich zorgen hoeft te maken bij een branch of de cpu nu wel of niet zal springen en waar naartoe. De status-analyser zal hem precies laten zien waarmee de cpu bezig is.
Het debuggen van zelfgemaakte programma's is nu een fluitje van een cent geworden (nauw ja, 1 cent, de complete print gaat FL.115,-- kosten, all-in)
Als ik deze JGS-tracer met status-analyser 9 maanden eerder had gebouwd dan was de euro-Atom al bijna weer ouderwets geworden.
Het is me nu wel duidelijk geworden dat ik een computersysteem ontworpen heb dat infeite meteen werkte .maar door een rare kronkel in de software, 9 maanden later pas operationeel werd.
Deze tracer met analyser heeft mij tot nu toe al meer plezier gegeven , dan de Atom in al die jaren en me ooit zal kunnen geven.
Eigelijk is deze print bedoeld voor iemand die systemen ontwerpt. In principe is deze analyser op ieder processor aan te sluiten. De pinbezetting van de cpu-connector en het Ready-of Halt-signaal moet veranderd worden. Het idee zal echter gelijk blijven.
Diegene die nu nog menen dat de hardware commisie niets doet zal waarschijnlijk bedoelen dat sommige figuren zich met zaken bezig houden waarmee de Atomist niets zal kunnen doen.
B.v Turbo op MDCR. tracers en statusanalysers waar geen hond wat mee kan doen.
Ik ben me dit terdege bewust en zal nu eerst eens het " ruisprobleem " op de Atom onder hand gaan nemen. Ik meende toch eerst de vooratgaand dingen te moeten realiseren omdat ikzelf hiermee meer belang heb. Je kunt beter een werkende Atom hebben die soms wat sneeuw op het scherm tevoorschijn tovert, dan een Atom die helemaal niets doet.

SCHEMA 6502 STATUS-ANALYSER.







Jan Swinkels
Pompstaat 41
6093 EA Heythuysen
04749 1235.

Programmering PROM's

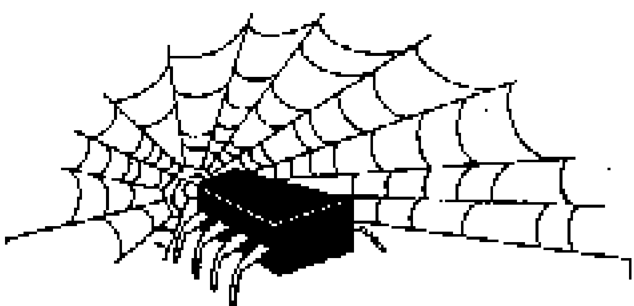
De PROM's zijn van het type 82S23. De 5 adres-ingangen zorgen dat er 8 data-uitgangen beschikbaar zijn. De PROM heeft open collector uitgangen, waardoor ze direct op een zeven-segment kunnen worden aangesloten. D.m.v. weerstanden wordt de stroom binnen de specificatie gehouden. De uitgang mag max. 20 mA leveren bij 1,7 Volt. De weerstand van 270 Ohm zorgt voor een stroombegrenzing van 5V/270 Ohm = 18,5 mA. We hebben maar 4 adres-ingangen nodig, vandaar dat we er een aan massa leggen (A5).Van de 8 uitgangen hebben we er maar 7 nodig voor het zeven-segment display. We gebruiken dus een uitgang (D7) gewoon niet.
Voor de aansturing van de segmenten dienen we er rekening mee te houden welk display we gebruiken, een met C.A of een met C.C.
De geprogrammeerde waardes moeten dan juist geinverteerd worden.Bij de PROM wordt indien de data "1" is, de uitgang hoog waardoor de led van het segment niet oplicht.
De tabel voor C.A display's staat hieronder afgebeeld.

adres	cijfer	segmenten aan	Data in PROM
		G F E D C B A	D 7 6 5 4 3 2 1 0
0000	0	F E D C B A	0 1 0 0 0 0 0 0
0001	1	C B	0 1 1 1 1 0 0 1
0010	2	G E D B A	0 0 1 0 0 1 0 0
0011	3	G D C B A	0 0 1 1 0 0 0 0
0100	4	G F C B	0 0 0 1 1 0 0 1
0101	5	G F D C A	0 0 0 1 0 0 1 0
0110	6	G F E D C A	0 0 0 0 0 0 1 0
0111	7	C B A	0 1 1 1 1 0 0 0
1000	8	G F E D C B A	0 0 0 0 0 0 0 0
1001	9	G F D C B A	0 0 0 1 0 0 0 0
1010	A	G F E C B A	0 0 0 0 1 0 0 0
1011	B	G F E D C	0 0 0 0 0 0 1 1
1100	C	F E D A	0 1 0 0 0 1 1 0
1101	D	G E D C B	0 0 1 0 0 0 0 1
1110	E	G F E D A	0 0 0 0 0 1 1 0
1111	F	G F E A	0 0 0 0 1 1 1 0

Helaas ben ik niet in staat om een printontwerp van deze tracer en status-analyser te maken. Met het plakken van printbanen en vervaardigen van printen ben ik reeds 6 jaar geleden mee gestopt. Het ontwerpen van schakelingen en die daarna met mijn 'roller-road pen" systeem werkt zo goed, dat ik definitief van het vervaardigen van printen, met de "ouderwetse" methode, ben afgestapt. Het gebeurt te vaak dat er een wijziging moet worden aangebracht. Bij geetste printen is dit nooit mogelijk op een fatsoenlijke manier. Het doorkrassen van printbanen vind ik zo'n "boerse en onbeholpen" manier dat ik hieraan niet mee wil helpen.
Zijn er personen die graag deze print willen hebben verzoek ik om zelf een print te maken. Het schema is door mij vrij gegeven. Het prototype, dat gebouwd is d.m.v. mijn "roller-road pen" systeem werkt zeer goed. Het schema mag daarom als foutief beschouwd worden.
Het is misschien wel verstandig om na te gaan wie er nog meer een print wil hebben, waardoor er maar een ontwerp gemaakt wordt en niet zoals bij de MDCR dat er 10 versie's zijn die allemaal iets anders "kunnen".
Bij de status-analyser heb ik gebruik gemaakt van PROM's die ik geprogrammeerd heb met de Electuur PROM-programmer.
Deze staat beschreven in Electuur juli/augustus 1980. Printnummer 80556.
Voor diegene die geen PROM-programmer bezitten wil ik wel de PROM's programmeren. Het is ook mogelijk om de bestaande IC's te gebruiken. Hiervoor wordt gebruik gemaakt van IC 9368. Dit is een BCD-Hex decoder. De werking van dit IC garandeer ik niet omdat ik met dit IC geen ervaring heb.
ik wens de Heren succes met het ontwerp van de print en alvast een plezierige vacantie, voor zover deze nog niet voorbij is.

Groetjes,

Jan Swinkels
Dorpsstraat 41
6093 EA Heythuysen
04749 - 1295



****
ELECTRON NIEUWS
****

Door Roland Leurs en Chris hilkman.

ATOM PRENTJES IN MODE 4

In de loop der jaren zijn in Atomnieuws heel wat olgard tekeningen gemaakt. Het is ook de moeite waard deze tekeningen om te zetten naar de Electron. Het programma hiervoor vindt u hieronder.
Alles wat u moet doen is de tekening inladen in de Atom, vervolgens met aecom overeturen naar de Electron (laden vanaf &2000 ! ! !) en de prent omzetten naar mode4.

Het conversieprogramma is in machinetaal geschreven, zodat alles lekker snel gaat. Na het omzetten wordt de tekening gecopieerd naar &2000. U kunt dan de tekening wegschrijven naar cassette of disk met:
 *SAVE PRENTJE 2000 47FF

Het tweede programma is een showprogramma. Het programma verwacht een cassette met tekeningen. Deze worden dan vanaf &2000 ingeladen. en na het laden op het scherm gezet.

Om aan te geven dat de cassette omgedraaid moet worden, moet er een z.g. STOP-file ingeladen worden. Deze file maakt u als volgt:
 $&2000="STOP"
 *SAVE "STOP" 2000 2005 Steeds als deze file wordt ingeladen maakt de Electron een sirene achtig geluid en wordt aangegeven dat de cassette omgedraaid moet worden.

Veel plezier ermee !

```
10 REM ATOM _ ELECTRON PICTURE CONVERSION
20 REM Written by :
30 REM Chris Hilkman - Urmond
40 REM Roland Leurs - Nattenhoven
50 FOR I%=0 TO 3 STEP 2
```

```
  60 P%=&C00
  70 [OPT I%
  80 .code   LDA #3 : STA &71 : LDA #&80 : STA  &72
\Y-coordinaat
  90         LDA #0 : STA &73 : LDA #&52 : STA  &74
\X-coordinaat
 100         LDA #0 : STA &75 : LDA #&20 : STA &76   \adres
Atom file
 110         LDA #0 : STA &78
 120         LDY #0
 130 .main   LDA(&75),Y : STA &77 : LDX #8
 140 .loop   ASL &77 : BCS plot
 150 .cont   JSR inc : LDA &78 : CMP #1 : BEQ rts
 160         DEX : CPX #0 : BNE loop
 170         INY : CPY #0 : BNE main
 180         INC &76 : JMP main
 190 .inc    CLC : LDA &74 : ADC #4 : STA &74
 200         LDA &73 : ADC #0 : STA &73
 210         CMP #4: BNE rts: LDA &74: CMP #&50 : BNE rts
 220         LDA #0 : STA &73 : LDA #&52 : STA &74
 230         SEC : LDA &72 : SBC #4 : STA &72
 240         LDA &71 : SBC #0 : STA &71
 250         CMP #0: BNE rts : LDA &72: CMP #&82: BCS rts
 260         LDA #1 : STA &78
 270 .rts    RTS
 280 .plot   LDA #25 : JSR &FFE3 : LDA #69 : JSR &FFE3
 290         LDA &74 : JSR &FFE3 : LDA &73 : JSR &FFE3
 300         LDA &72 : JSR &FFE3 : LDA &71 : JSR &FFE3
 310         JMP cont
 320 ]
 330 NEXT I%
 340 MODE4:VDU 23,1,0;0;0;0;:CALL code
 350 FOR X%=0 TO &27FF STEP 4:X%!&2000=X%!&5800:NEXT X%
 360 PRINT "De tekening staat van &2000 - &47FF."
 370 END
```

TEKENING PARADE :

```
  10 REM ***************************
  20 REM *                         *
  30 REM * AUTOMATIC PICTURE SHOW  *
  40 REM *                         *
  50 REM *    program by R. Leurs  *
  60 REM *                         *
  70 REM *    (c) 1989 SMURFSOFT   *
  80 REM *                         *
  90 REM ***************************
 100
```

```
  110 FOR I%=0 TO 2 STEP 2:P%=&1F00
  120 [OPT I%:.COPY
  130 LDA #&20:STA &71:LDA #&58:STA &73
  140 LDA #&00:STA &70:STA &72:LDX #&27
  150 .L1 LDY #&00:.L2 LDA (&70),Y
  160 STA (&72),Y:INY:CPY #&00:BNE L2
  170 INC &71:INC &73:DEX:BPL L1:RTS
  180 ]:NEXT I%
  190
  200 *OPT
  210 MODE 4:VDU23,1,0;0;0;0;:PROCscroll("SMURFSOFT
PRESENTEERT  ELECTRON TEKENING PARADE")
  220 MODE 5:VDU 19,3,0;0;
  230 PRINT "Dit programma laadt"''"de tekeningen
vanaf"''"cassette in en"''"toont deze op het
"''"beeldscherm"''
  240 PRINT"Als er een fout op-"''"treedt zal
het"''"programma het laden"''"niet afbreken; een"
  250 PRINT'"fout zal te zien zyn"'"als een fout in
de"''"tekening."''':COLOUR2:PRINT"Druk op de
spatiebalk";:VDU19,3,7;0;:REPEAT UNTIL GET=32
  260 MODE 4:VDU 23,1,0;0;0;0;
  270 PRINT '''"CASSETTE DATA IN BEELD  J/N ?":REPEAT
W%=GET:UNTIL W%=ASC"J" OR W%=ASC"N"
  280 VDU12,28,10,2,30,0
  290 IF W%=ASC"N" THEN *OPT 1,0
  300 *OPT 2,0
  310
  320 REM LOAD AND COPY
  330 ON ERROR GOTO 430
  340 REPEAT
  350 *LOAD "" 2000
  360 IF $&2000="STOP" THEN GOSUB 490:GOTO 410
  370 VDU 19,1,0;0;
  380 CLG
  390 CALL COPY
  400 VDU 19,1,7;0;
  410 UNTIL FALSE
  420
  430 VDU 26,12
  440 ON ERROR OFF
  450 PROCscroll("TOT ZIENS")
  460 MODE6:PRINT'"Acorn Electron"''"BASIC"'
  470 END
  480
  490 PRINTCHR$12"CASSETTE";CHR$10;"OMDRAAIEN";CHR$10;"SVP";
  500 SOUND 1,2,100,100:ENVELOPE 2,1,4,-4,4,10,20,10,126,0,0
,-126,126,126
```

```
510 SOUND 1,2,100,100:SOUND 1,2,100,100
520 NU=TIME:REPEAT UNTIL TIME-NU>1500
530 CLS:RETURN
540
550 DEFPROCscroll(t$)
560 VDU 23,255,255,255,255,255,255,255,255,255
570 A%=10:X%=&70:Y%=0:REPEAT ?&70=ASC(t$):CALL &FFF1
580 FOR J%=0 TO 7:a$="":FOR K%=7 TO 0 STEP -1
590 IF 2^K% AND J%?&71 a$=a$+CHR$255 ELSE a$=a$+" "
600 NEXT:PRINTTAB(14,31);a$:NEXT:t$=MID$(t$,2):UNTIL t$=""
610 ENDPROC
```

D I V E R S E   Z A K E N

door Henk Bastings

1. WAT MOET IK NU MET WORTELTREKKEN.

Met dit stukje programmatuur (WORTELT) wil ik alleen een routine laten zien die in staat is de wortel uit een getal te trekken, zonder gebruik te maken van de routine die immers in de floating-point rom aanwezig is. Veel rekenmachine's berekenen op deze manier namelijk de wortel uit een getal.
De nauwkeurigheid waarmee dit gebeurt hangt af van de precieze waarmee een F.P. getal kan worden verwerkt. In dit geval is deze dus precies gelijk aan de nauwkeurigheid van de ATOM.
Een gek zou er dus nog iets beters van kunnen maken !.
Als benadering mag je elk getal invullen en hoe nauwkeuriger je dit doet, des te sneller komt het eindresultaat in zicht.

2. SPATIES VERWIJDEREN.

REMOVSP is een afkorting van remove space en dient in dit geval nu eens een keer voor het verwijderen van spaties aan het einde van de regel, die waarschijnlijk ontstaan zijn door het te ver copieren van een regel of iets dergelijks.
Aangezien deze spaties dan meestal alleen dienen als extra verlenging zonder enig doel, kun je deze spaties dus ontmissen als kiesprijs.
REMOVSP zoekt deze spaties voor je op en verwijdert ze netjes.
Het programma verkort alleen een bestaand programma en heeft dan ook geen extra geheugen nodig. Verder is het weer geschreven in standaard basic en iets minder snel, hi hi.

```
*****************************
*   VDU 3.4 80-kolomssoft   *
*****************************
```

door Peter Wokke

Alweer een nieuwere versie. ja dat wel maar niet zonder reden, want door een gigantisch grote blunder van mijn kant is VDU-3.3 weliswaar de kortste maar niet de perfectste, want er zijn diverse bugs ingeslopen. Ook de kritiek die in de afgelopen nummers naar voren gekomen is, was voor mij een reden om het een en ander aan te passen en/of te veranderen. Dat is dus de reden dat VDU 3.4 geboren is, een nieuwe telg in de VDU familie, met nog grotere haast ongekende mogelijkheden. Helaas is de soft ook weer iets groter geworden (vdu + ext ongeveer 3Kb). Maar de voordelen wegen niet op tegen de geheugenruimte die nodig is.

De nieuwste uitbreidingen zijn, HIGH- en LOW-VIDEO in 80 kolomsmode dit in navolging van de 80-koloms elektuur-kaart (zie A.N.7.4 pag 27-28) maar dan ZONDER hardware wijzigingen. Dit op veler verzoek van de CPM-kaart gebruikers die deze mogelijkheid tot nu toe moesten missen in atommode. Omschakelen tussen high en low video gaat met behulp van CTRL A (low-video) en CRTL-Z (high-video).

SCREEN is de tweede vernieuwing, hiermee kan je op (maximaal) 4 schermen onafhankelijk van elkaar werken. Dit is afhankelijk van de hoeveelheid geheugen op de 80 kolomskaart. Als regel mag je aannemen 4 Kilobyte per scherm, dus een volle kaart (16 Kb) herbergt 4 schermen.

PON en POFF zijn de commando's voor pagemode (P.$14/P.$15). Als de pagemode aan staat stop het scrollen onderaan het scherm en gaan na een toetsaanslag op een leeg scherm door.

INSCREEN en OUTSCREEN haalt of zet een compleet scherm in of uit het interne geheugen. OUTSCREEN #xxxx haalt een scherm uit het interne geheugen en plaatst deze in het atom-geheugen vanaf #xxxx.(een scherm gebruikt 3 Kb).INSCREEN #xxxx zet een scherm in het interne geheugen van de 80-kolomskaart.

KARSET is een uitbreiding die de UDS karakterset aan zet. De karakters 32 t/m 127 worden verwisseld door de UDS karakters 160 t/m 255, mits een UDS blok aanwezig is natuurlijk. Een voorbeeld vindt je op de regio/clubschijf. Run het programma maar eens en type vervolgens KARSET.

INUDS en OUTUDS zijn de laatste uitbreidingen, hiermee is het mogelijk om een heel blok met eigen gemaakte UDS-karakters in/uit de 80-kolomskaart te zetten/halen en in het normale atoomgeheugen te plaatsen. Een blok UDS karakters gebruikt 1 Kb geheugenruimte.

Ook de andere Extenties hebben ook een face-lift gehad en werken nu volgens de regels,d.w.z. ze springen niet meer in de VDU-soft, dus ze zijn overal te plaatsen. (VDU-3.4 moet natuurlijk wel geinstalleerd zijn). Uitleg over alle extenties vind je op de volgende pagina's.

LET OP !!!

Met het statement VDU kan de soft niet worden opgestart. Dit komt doordat VDU 40,x of VDU 80,x eerst screen x uitvoert, vervolgens het linebyte (#EA) op 80 resp. 40 kolommen instelt en tot slot een restart (=P.#1B) uitvoert. Er wordt dus niet gesprongen naar de VDU-soft zelf. De reden hiervan is dat bij bijv. VDU 40,2 het eerste screen in 40 kolomsmode niet veranderd wordt, hetgeen wel gebeurt bij het aanroepen van soft. Dit aanroepen doet U met LINK #7000 (bij VE-3.4).

Het kan natuurlijk ook zijn dat u het hier niet mee eens bent. Verander dan in het programma EXT-3.4 de regels 4270 en 4290:

```
    4270    JMP #7B00  \ het beginadres 40 kol.mode
    4290    JMP #7D04  \ het beginadres 40 kol.mode
```

Na gebruik van VDU 40 of VDU 80 komt u wel altijd in screen 1. Met deze wijziging in de 80 kolomssoft wel aan te roepen vanuit welke VDU soft dan ook. Een ander alternatief is zelf er een F-CHARME statement bij te maken. Wijzig dan EXT-3.4 volgens onderstaande methode:

```
      40 IN."WAAR EXT.TABEL    ( #7300) "T
      50 IN."WAAR EXT.ASSEM    ( #7440) "A
      70 DIM LL79,XX15,ZZ36
      80 F.X=0 TO 132;LL.X=-1;N.
     861 $T="INITVDU";T=T+LEN(T)
     862 ?T=ZZ36/256|#80;T?1=ZZ36%256;T=T+2
   4391:ZZ36 JSR XX0;\           INITVDU A
   4392      JSR LL78;JMP #C55B
   4393:LL78 CMP @#1F;BNE LL79
   4394      JMP #7B00;\ het beginadres 40 kol.mode
   4395:LL79 CMP @#9F;BNE LL75
   4396      JMP #7B04;\ het beginadres 80 kol.mode
```

Op de clubschijf staan de volgende programma's:

VDU-3.4	2900	C2B2	5D66	34AA	source van VDU 3.4
EXT-3.4	2900	C2B2	56D5	2DB5	source van EXT 3.4
VE 3.4	7400	7F90	7FAF	00AF	geassembleerde soft
VDU-UDG	2900	C2BV	3EDB	1500	UDG karaktergenerator

Zijn er nog vragen, suggesties en/of opmerkingen bel me dan.

Peter Wokke

UITLEG EXTENTIONS 3.4

Op de volgende acht pagina's vindt U de beschrijving van de mogelijkheden van alle statements uit EXT-3.4.

COMMANDS

Aanroep: COMMANDS
Mode: 40 - 80
Uitvoering: Laat alle extensions op het scherm zien

COMMANDS	KADER	SEMIGRAPH	ACCENT
COLOR	FRGNDCOLOR	BACKCOLOR	MARGIN
CURS	CLS	PON	POFF
GRAPHIC	SERVICE	RESTART	NEGATIVE
DWIDTH	DHEIGTH	FLASH	UNDERLINE
SPECIAL	CLC	LOCATE	CURPOS
VERSION	DDS	DEL	SCREEN
VDU	LOWVIDEO	HIGHVIDEO	OUTUDS
INUDS	OUTSCREEN	INSCREEN	KARSET

KADER

Aanroep: KADER A,B
Mode: 40
Uitvoering: Tekent een kader op het scherm met lengte A en hoogte B

SEMIGRAPH

Aanroep: SEMIGRAPH A,x,x,x....
Mode: 40
Uitvoering: Tekent een subgrafisch karakter A op het scherm het commando is repeterend uitgevoerd d.w.z SEMIGRAPH A,A,A,A,A,A tekent 6 karakters A
A is een getal tussen 0 en 31
Voorbeeld: SEMIGRAPH 10 (is #0A & #1F -> #0A)
SEMIGRAPH 42 (is #2A & #1F => #0A)

ACCENT A,B

Aanroep: ACCENT A.B
Mode: 40
Uitvoering: Tekent een karakter A met accent B op het scherm.
A is een getal tussen #20 en #50
B is een getal tussen 0 en 8

COLOR

Aanroep: COLOR A,B,C
Mode: 40 80
Uitvoering: Veranderd de kleur met A als voorgrondkleur, B als achtergrondkleur en C als randkleur (C is alleen mogelijk in 40 kolomsmode)
A,B en C zijn getallen tussen 0 en 7

FRONTCOLOR

Aanroep: FRONTCOLOR A
Mode: 40 - 80
Uitvoering: Veranderd de voorgrond kleur (de karakters)
A is een getal tussen 0 en 7
Voorbeeld: FRONTCOLOR 6 (#06 & #07 => #06)
FRONTCOLOR 50 (#32 & #07 => #02)

BACKCOLOR

Aanroep: BACKCOLOR A
Mode: 40 - 80
Uitvoering: Veranderd de achtergrondkleur
A is een getal tussen 0 en 7

MARGIN

Aanroep: MARGIN A
Mode: 40
Uitvoering: Veranderd de randkleur
A is een getal tussen 0 en 7

CURS

Aanroep: CURS A
Mode: 40 - 80
Uitvoering: Verander de cursor in blok/streep/vast/knipperend
A is een getal tussen 0 en 4
A=0 cursor uit
A=1 cursor aan blok vast
A=2 cursor aan blok knipperend
A=3 cursor aan streep vast
A=4 cursor aan streep knipperend

CLS

Aanroep: CLS
Mode: 40 - 80
Uitvoering: Maakt het scherm schoon
Ctrlcode: P.#12
Ctrltoets: CTRL - L
CTRL .

PGN

Aanroep: PGN
Mode: 40 - 80
Uitvoering: Pagemode aan, scrollen stop onderaan het scherm en
gaat na toetsaanslag bovenaan het scherm weer door
Ctrlcode: P.$14
Ctrltoets: CTRL - N

```
POFF

Aanroep:      POFF
Mode:         40 - 80
Uitvoering:   Pagemode uit, scrollen gaat gewoon door
Ctrlcode:     P.#15
Ctrltoets:    CTRL - O

GRAPHIC
-------
Aanroep:      GRAPHIC
Mode:         40
Uitvoering:   Schakelt grafische karakterset in of uit
Ctrlcode:     P.#16
Ctrltoets:    CTRL   0
              CTRL - P

SERVICE
-------
Aanroep:      SERVICE ".....tekst....."
Mode:         40 - 80
Uitvoering:   Beschrijft de service regel, zonder "tekst"
              word deze leeg gemaakt
Ctrlcode:     P.#17
Ctrltoets:    CTRL - 1
              CTRL - Q

RESTART
-------
Aanroep:      RESTART
Mode:         40 - 80
Uitvoering:   Start opnieuw de vdu routine, maakt de service
              line en het scherm leeg, maar blijft in dezelfde
              schermmode [?]
Ctrlcode:     P.#18
Ctrltoets:    CTRL - 2
              CTRL - R

NEGATIVE
--------
Aanroep:      NEGATIVE
Mode:         40 - 80
Uitvoering:   Zet invertermode aan of uit
Ctrlcode:     P.#17
Ctrltoets:    CTRL - 3
              CTRL - S

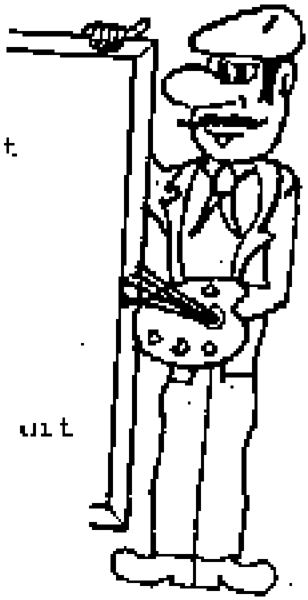
DWIDTH
.
Aanroep:      DWIDTH
Mode:         40
Uitvoering:   Zet de dubbelbreed mode aan of uit
Ctrlcode:     P.#20
Ctrltoets:    CTRL - 4
              CTRL - T
```

DHEIGTH

Aanroep: DHEIGTH
Mode: 40
Uitvoering: Zet dubbelhoog modes aan of uit, alle regels moeten twee keer onder elkaar geschreven worden
Ctrlcode: P.$22
Ctrltoets: CTRL - 6
CTRL - V

FLASH

Aanroep: FLASH
Mode: 40 - 80
Uitvoering: Zet het knipperen aan of uit
Ctrlcode: P.$23
Ctrltoets: CTRL - 7
CTRL - W



UNDERLINE

Aanroep: UNDERLINE
Mode: 40 - 80
Uitvoering: Zet het onderstrepen aan of uit
Ctrlcode: P.$24
Ctrltoets: CTRL - 8

SPECIAL

Aanroep: SPECIAL
Mode: 40 - 80
Uitvoering: Zet de speciale karakters in de karakterset aan of uit
Ctrlcode: P.$25
Ctrltoets: CTRL - 9
CTRL - Y

CLC

Aanroep: CLC
Mode: 40 - 80
Uitvoering: Maakt het scherm schoon vanaf de cursor

LOCATE

Aanroep: LOCATE A,B
Mode: 40 - 80
Uitvoering: Zet de cursor op positie A in regel B

CURPOS

Aanroep: CURPOS A,B
Mode: 40-80
Uitvoering: Haalt de cursor positie op en zet deze in de variabelen A en B

VERSION

Aanroep: VERSION
Mode: 40 80
Uitvoering: Geeft versie nummer van de VDU soft

UDS

Aanroep: UDS A,x,x,x,...
Mode: 40
Uitvoering: Zet eigen uds-karakter A op het scherm
het commando is repeterend uitgevoerd d.w.z
UDS A,A,A,A,A,A tekent 6 uds-karakters A
A is een getal tussen 160 en 255, getallen kleiner
dan 128 worden normaal uitgevoerd.
Voorbeeld: UDS 160,37,8,7,9,45,...

DCT

Aanroep: DCT A,t1,t2,t3,t4,t5,t6,t7,t8,t9,t10
Mode: 40
Uitvoering: Vult uds-karakter A met de waarde t1 t/m t10
A is een getal tussen 160 en 255
Voorbeeld: DCT 160,#FF,#81.#81,#81,#81,#81,#81,#FF,#00,#00
maakt een open vierkantje

SCREEN

Aanroep: SCREEN A
Mode: 40-80
Uitvoering: Schakelt over naar scherm A maar veranderd niets
in het oude en nieuwe scherm
A is een getal tussen 1 en 4 (afhankelijk van
het interne geheugen 4K6 per scherm)

VDU

Aanroep: VDU A,B
Mode: 40 - 80
Uitvoering: Gaat naar scherm B in A kolomsmode, (als B niet
gegeven is screen 1)
A = 32, 24, 40, 80 mode
B = 1 - 4 (zie screen)
Voorbeeld: VDU 40 40 kolomsmode screen 1
VDU 80,3 80 kolomsmode screen 3
P.S: Dit werkt alleen als de 80 kolomskaart al eerder
aangeroepen is.(zie artikel VDU-3.4 in dit blad)

LOWVIDEO

Aanroep: LOWVIDEO
Mode: 80
Uitvoering: Zet dim-mode aan (als frontcolor 5 is is er GEEN
verschil)
Ctrl codes: P.$1
Ctrltoets: CTRL

HIGHVIDEO

```
Aanroep:      HIGHVIDEO
Mode:         80
Uitvoering:   Zet dim-mode uit
Ctrlcode:     P.#26
Ctrltoets:    CTRL - Z
```

KARSET

```
Aanroep:      KARSET
Mode:         40
Uitvoering:   Zet UDS karakterset aan of uit
              karaktercodes 32 - 127 worden verwisseld met de
              UDS karakters 160 - 255 (mits UDS blok aanwezig)
Voorbeeld:    F.Q=32 TO 255;P.$Q;N.Q;\ normaal
              KARSET;F.Q=32 TO 255;P.$Q;N.Q;KARSET;\ verwisseld
```

OUTUDS

```
Aanroep:      OUTUDS A
Mode:         40 - 80
Uitvoering:   Haalt UDS - karakterblok op en zet deze in het
              normale ATOM geheugen vanaf A (#....)
              Dit blok is #3BF bytes groot (96 karakters x 10)
Voorbeeld:    OUTUDS #4000
              *SAVE UDSBLOK 4000 43BF
```

INUDS

```
Aanroep:      INUDS A
Mode:         40
Uitvoering:   Haalt UDS - karakterblok uit het normale ATOM
              geheugen vanaf A (#....) en zet deze in het
              geheugen van de 80-kolomskaart (#3BF bytes)
Voorbeeld:    *LOAD UDSBLOK 4000
              INUDS #4000
```

OUTSCREEN

```
Aanroep:      OUTSCREEN A
Mode:         40 - 80
Uitvoering:   Haalt een compleet scherm op en zet deze in het
              geheugen vanaf A (#....) een scherm gebruikt
              #B3F bytes ( 40 x 24 x 3  of  80 x 24 x 1.5)
Voorbeeld:    OUTSCREEN #4000
              *SAVE SCRN80 4000 4B3F
```

INSCREEN

```
Aanroep:      INSCREEN A
Mode:         40 - 80
Uitvoering:   Haalt een scherm uit het ATOM geheugen vanaf A
              en plaatst deze in de 80 kolomskaart
Voorbeeld:    *LOAD SCRN80 4000
              INSCREEN #4000
```

INDELING ZEROPAGE en ATRIBUTEN

Zie voor verdere info datasheet EF9345 en A.N.8-2 blz 13 & 14

40 KOLOMSMODE

ATRIBUUT A #DE

bit 7	negatief
bit 6,5,4	voorgrondkleur
bit 3	knipperen
bit 2,1,0	achtergrondkleur

ATRIBUUT B #9FE2

bit 7,6,5,4	type & mod
bit 3	dubbelbreed
bit 2	verborgen (nog niet toegepast)
bit 1	dubbelhoog
bit 0	insert (nog niet toegepast)

STATUSBYTE #DF

bit 7	scherm uit
bit 6	speciale karakters
bit 5,4,3,2	schermwijze
bit 1,0	teller escape

CURSORBYTE #E0

bit 7	pagemode
bit 6,5,4,3,2,1,0	teller cursorpositie

MATBYTE #9FE3

bit 7	dubbelhoog
bit 6,5,4	cursor (blok/streep/vast/knipper)
bit 3	insert margin (nog niet toegepast)
bit 2,1,0	rand kleur

LINEBYTE #E6

bit 7	40/80 kolom
bit 6	UDS aanwezig
bit 5	karakterset normaal/UDS
bit 4,3,2,1,0	maximaal aantal regels op scherm

80 KOLOMSMODE

ATRIBUUT A #7FE2

bit 7	negatief even positie
bit 6	flash even positie
bit 5	onderstrepen even positie
bit 4	high/low video even positie
bit 3	negatief odd positie
bit 2	flash odd positie
bit 1	onderstrepen odd positie
bit 0	high/low video odd positie

DORREGISTER #DE

bit 7	insert even (nog niet toegepast)
bit 6,5,4	voorgrondkleur lowvideo
bit 3	insert even (nog niet toegepast)
bit 2,1,0	voorgrondkleur highvideo

STATUSBYTE #DF

bit 7	scherm uit
bit 6	speciale karakters
bit 5,4.3,2	schermkeuze
bit 1,0	teller escape

CURSORBYTE #E0

bit 7	pagemode
bit 6,5,4,3,2,1,0	teller cursorpositie

MODBYTE #9FE3

bit 7	dubbelhoog
bit 6.5,4	cursor (blok/streep/vast/knipper)
bit 3	insert margin (nog niet toegepast)
bit 2,1,0	achtergrond kleur

LINEBYTE #EA

bit 7	80/40 kolom.
bit 6	-
bit 5	-
bit 4,3,2,1,0	maximaal aantal regels op scherm

o.s. de keuze van sommige adressen (#9FEx) zijn persoonlijke voorkeur en kunnen desgewenst veranderd worden
(ook in de extentions !!)

Peter Wokke
Dukaatstraat 12, 1827 RF ALKMAAR tel:072 - 621135

```
*********************
*    UDS - EDITOR   *
*********************
```

door Peter Wokke

Sinds het mogelijk is om UDS karakters te gebruiken, was je verplicht om deze karakters stuk voor stuk te tekenen, te berekenen en/of te definieren met het DCT-kommando. Bij de laatste VDU-soft (versie 3.4) zitten een aantal extra kommando's die speciaal voor de UDS-karakterset geschreven zijn. Bijvoorbeeld INUDS, OUTUDS en KARSET zijn handige commando's die het je wel heel makkelijk maken om met die karakterset te werken. Met de hier beschreven editor is het maken van een karakterset ook vrij simpel geworden. Met ander woorden iedereen kan nu zijn eigen karakters maken. Het wachten is nu op de eerste toepassing hiervan.

Omdat de editor geheel in (P-Charme) basic geschreven is duren enkele handelingen wat lang, maar er is goed mee te werken en niet minder belangrijk voor iedereen leesbaar. Na het opstarten krijg je een scherm dat in vier stukken verdeeld is. Bovenin het beeld staan de toetsen met hun werking vermeld. onderin staat de code van het karakter voor het DCT kommando en in het midden staan twee velden. Het linker veld geeft een overzicht van alle karakters en rechts staat het aangewezen UDS-karakter.

De werking van de EDITOR is hieronder beknopt weergegeven, zie verder de aanwijzingen op het scherm. Laad de UDS-EDITOR en laad een blok UDS-karakters, en run de EDITOR. Wijs met de toetsen A, Z,< en > het te maken karakter aan op het linkerveld. Druk op E(dit). Maak of verander het karakter op het rechterveld met de toetsen A, Z, <, >. S(et/ reset) en I(nvert) of C(lear). Druk op Q(uit). Beantwoord de vragen UDS OKE en PLACE OKE (geef nieuwe positie op). Ga door met nieuw karakter of stop (met N). Als je stop vergeet het blok niet te saven met de onderste schermregel.

Als extra zitten al een paar UDS blokken bij met allerlei voorbeelden en mogelijkheden.

UDS ED	2900	C2B2	3C0D	130D	UDS editor/maker
UDS k.1	5000	0000	530F	03BF	blok UDS karakters
UDS-k.2	5000	0000	530F	03BF	blok UDS karakters
UDS-k.3	5000	0000	530F	03BF	blok UDS karakters

Peter Wokke

DIGITIZERITIS.

Al weer bijna twee jaar geleden hebben we eens een regio middag gehad tezamen met de regio Delft. Op die middag waren toen onder meer de toen nog in een prematuur stadium verkerende, maar perfect werkende, Z80-CP/M kaart te zien en ook een digitizer. Deze laatste interesseerde me mateloos en mijn grote wens was dan ook zelf zo'n ding te bouwen.
Bij navraag of er ook een print voor te krijgen was bleek het antwoord negatief. Je moet dat gewoon even doorlussen zei men. Nu heb ik weleens zo'n doorgelust geval gezien, maar dat leek me echt niks. De bovenkant zag er nog wel uit, maar bij de onderkant vergeleken was een bord spagetti een ordelijk georganiseerd geheel. Plan om te bouwen dus voorlopig op de plank.
Een flinke tijd later leerde ik hoe het afwrijven van printjes in z'n werk ging. Na wat kleiner probeerwerk kwam het digitizerplan weer van de plank. Na een hoop gepuzzel op ruitjes papier omtrent, plaatsing van IC's en allerlei verschillende sporenplannen, kwam er een proefprint uit de bus Na ontwikkelen etsen boren en bouwen bleek het geheel nog redelijk te werken ook, maar er zaten nog wel en paar fouten in. Zo had ik in m'n enthousiasme de 64-pins steker verkeerd om op de print getekend. Dom, maar niet onoverkomelijk. Verder bleek de instelling van het contrast en de helderheid nogal kritisch. Op de regioavond waar ik het ding demonstreerde vertelde iemand, die wel wat van elektronica weet, me dat de schakeling rond de TBA 790 stond te oscilleren. Dat probleem is met een dikke C opgelost. Verder bleek dat een goede ontkoppeling de werking belangrijk verbetert.
De laatste tijd ben ik bezig geweest de aanwezige fouten op de kaart weg te werken en er ligt nu een print klaar die zonder grote problemen zou moeten werken.
Op de kaart bevinden zich de digitizer, de uitdecodering naar #BFD4 tot #BFD7 en een schakeling voor de opwekking van + en - 12 volt. Het verbruik ligt zo rond de 700 mA met ingeschakelde 12V opwekking. Haalt u die 12V van uw netvoeding, dan loopt het verbruik terug naar zo'n 350 mA. Zoveel mogelijk zijn de gebruikte IC's voorzien van een ontkoppel condensator. Waar dit moeilijk of niet te realiseren viel moet u, indien de werking instabiel is, zelf wat extra ontkopel C's aanbrengen. Zeker de 4 LM 319'e willen graag voorzien worden van ontkoppelaars. Deze moeten of op het ictje, of onder op de print gesoldeerd worden.
De bouw:
Als je besluit deze kaart te gaan bouwen neem er dan vooral de tijd voor. Als eerste moeten alle draadbruggen aangebracht worden. Doe dit vooral voordat u de voeten aanbrengt, want veel verbindingen lopen alleen maar onder een voetje of lopen onder meerdere voeten door of hebben hun aansluitpunten onder een voet. Om de aansluit punten van de draden gemakkelijker te kunnen vinden is het wel eenvoudiger om de voeten even los op hun plek in de gaatjes te zetten.
Als alle draadbruggen op hun plek zitten kunnen de voeten erin. Als die zitten ga dan eerst met het schema ernaast eens na of

alle verbindingen er zijn. Dus dat er bv geen breuken in spoortjes en of kortsluitinkjes tussen spoortjes voorkomen door kloddertjes soldeer of zo. De spoortjes lopen hier en daar namelijk zeer dicht langs elkaar.
Als u de 12 V van de kaart wilt halen is het verstandig de schakeling die deze opwekt eerst te bouwen en te testen. zodat u er zeker van bent dat er niet ineens veel meer dan 12 V op de IC's komt te staan. Als test belasting kunt u een weerstand van pakweg 100 tot 110 ohm gebruiken voor zowel de + als de - 12 V. Zet tijdens het afregelen van de schakeling wel even en koelplaatje op de BD139 omdat wanneer u de nog niet afgeregelde schakeling opstart en de potmeter in een ongunstige stand staat de BD139 binnen een paar tellen bloedheet wordt en als het even meezit met een diepe zucht de geest geeft. Is de zaak eenmaal juist ingesteld, dan is er waarschijnlijk niet eens een koelplaat meer nodig. Bij mijn proef kaart is ie na uren gebruik niet eens handwarm. De potmeter moet u zo instellen dat het stroom verbruik zo laag mogelijk is, terwijl de spanning juist de gewenste waarde bereikt (met de belastings weerstanden aangesloten). Als het zaakje goed is ingesteld hoeft het totale stroomverbruik niet hoger te zijn dan ongeveer 700 mA. De BD139 kan het dan waarschijnlijk zonder koelplaatje af. Zit de uitgangs spanning wat aan de hoge kant dan is er voorzien in een mogelijkheid om een spanningsstabilisator op de kaart aan te brengen. In de meeste gevallen zal dit echter niet nodig zijn.
Het aantal wikkelingen op het spoeltje bedraagt 100 en de draaddikte is 0,2 mm. Het spoellichaam van ferroxcube meet 35 mm bij een diamete van 6 mm.
Nu kunnen alle weerstanden condensatoren enz. erop. voor de condnsatoren in de digitizer moeten tantaal types gebruikt worden. in de schakeling rond de TBA 790 staat C10 aangegeven voor 0.68 uF. Dit moet zijn 68 uF (volgens de gegevens van de datasheet).

Gebruik voor het bouwen een goede soldeerbout met een fijn puntje om overvloeien van soldeermatriaal tussen de spoortjes te voorkomen.
Problemen met de synchronisatie heb ik bij geen van beide gebouwde digitizers gehad.
Vlak naast de TBA790 is er plaats voor een weerstand van 2k. Deze staat wel aangegeven op de datasheet maar niet in het schema. Bij mij maakte het niet uit of ie er wel of niet inzat. Bij de in het schema voorkomende potmeters wordt door de M het lopercontact aangeduid.
Verdere inlichtingen over het gebruik van de kaart en de werking kunt u vinden in Atom Nieuws, de jaargang 5 de nrs5 en 6 en jaargang 6 de nrs 2 en 4.

Hier volgt nog even de onderdelen lijst.

R1 - 100 K	R2,4,5 - 1K	R3,6,10,28 - 12K
R7 - 1,5 K	R8 - 27K	R9,12 -2,7 K
R11- 1,2 K	R13 - 750 ohm	R14,15,16,17,18,19 - 150 ohm
R20- 1,8K	R21,22,23,24,25,26,27 - 470 ohm	
R29- 10K R30-10K		
C1 -4,7 uF	C2,14,15 - 1nF	C3,9 - 1 uF
C4,5 - 100 nF	C6 - 100 pF	C7,8,12,13 - 10 uF
C10 - 68 uF	C11- 120 pF	
Z1 - 5,6 V	T1 - BC309	T2,3,4 - BC238
P1 - 5K	P2,4 - 10K	P3 - 50K
IC1 - 78105	IC2 - 74LS629	IC3 - 74LS132
IC4 - 74LS395	IC5 - 74LS395	IC6 - 74LS395
IC7 - 74LS395	IC8 - 74LS76	IC9 - 74LS193
IC10- 74LS193	IC11- 74LS85	IC12- 8255
IC13- 74LS151	IC14- 74LS148	IC15- 74LS14
IC16-LM319	IC17- LM319	IC18- LM319
IC19- LM319	IC20- TBA970	IC21- 74LS139
IC22- 74LS123	ICa1- 74LS133	ICa2- 74LS27

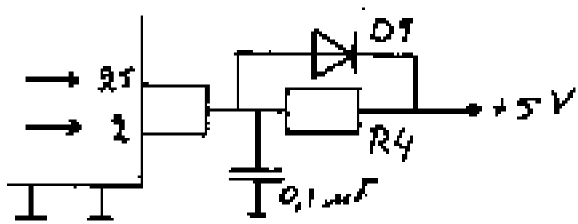
Ik hoop dat de print te zijner tijd opgenomen zal worden in de Sanders Print Service.
Ik wil vanaf deze plaats iedereen die me met raad en daad geholpen heeft dit projectje tot een goed einde te brengen van harte bedanken, evenals de ontwerper van dit moois. Eventuele bouwers wens ik veel succes en plezier met hun digitizer.

Theo Waaijer

Voor het bestellen van de print eerst even telefonisch contact opnemen met E. Sanders. Tel. zie SPS elders in dit blad.

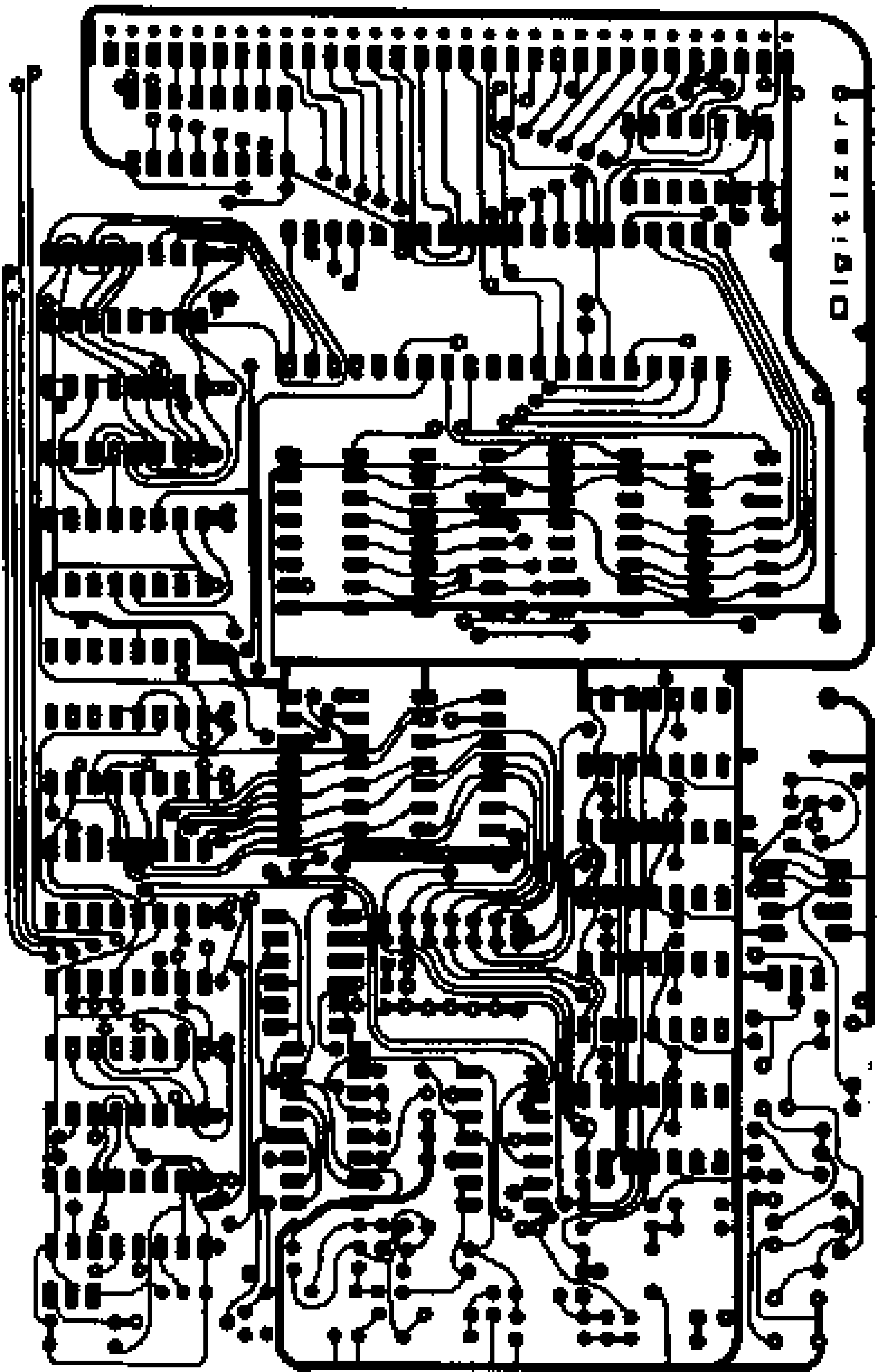
F O U T W A R E

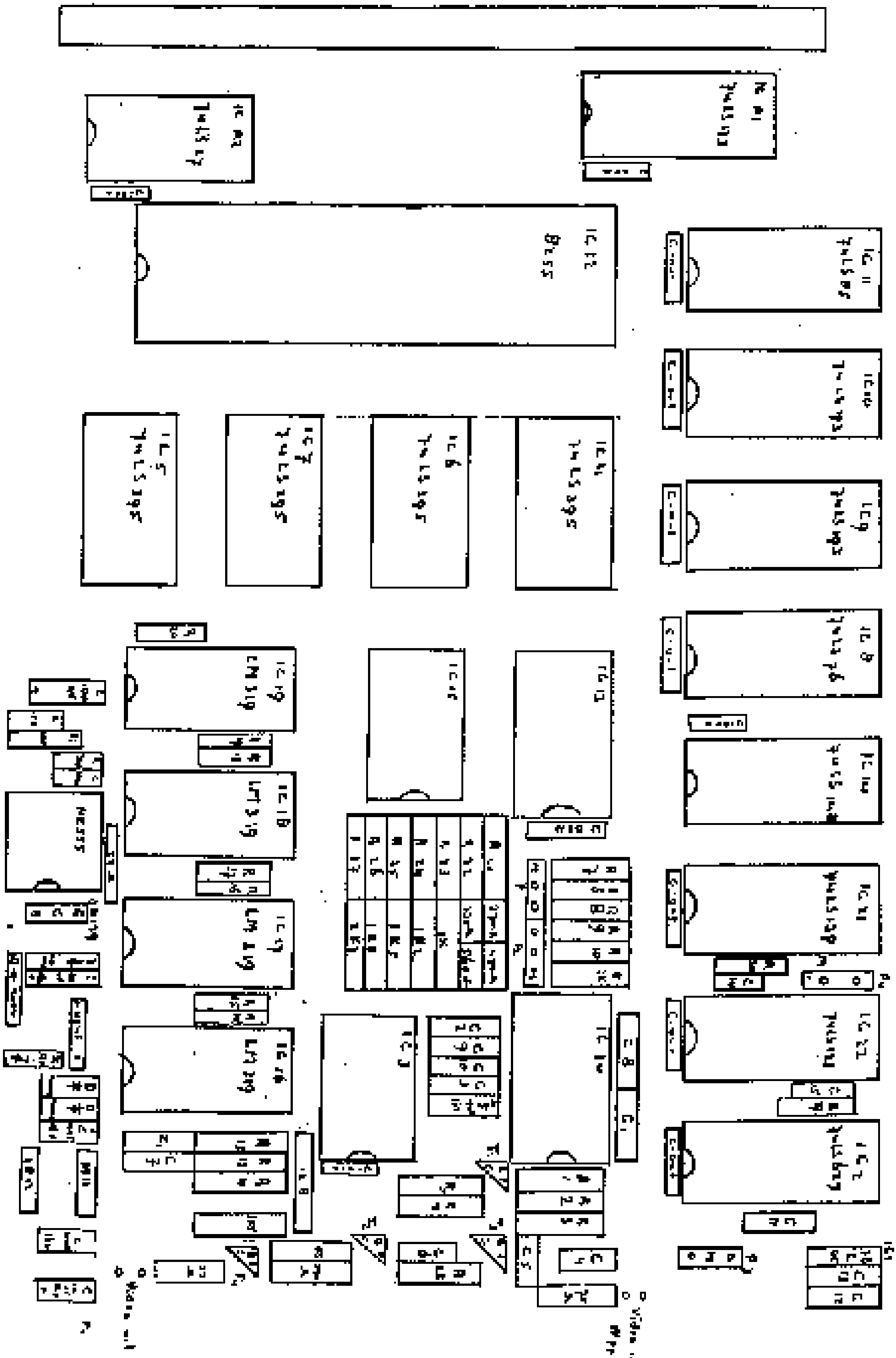
Er ontbreekt iets aan het schema van de Atom Speech Synthesizer in AN89-2, blz 8. Als u het schema bekijkt valt u misschien op dat er rechtsonder aan de SP0256-al2 twee pinnummers niet vermeld staan. De ontbrekende pinnummers zijn 25 en 3. Zie ook onderstaande figuur. Als u deze twee nummers even bijschrijft op bovengenoemde bladzijde, dan klopt alles weer!

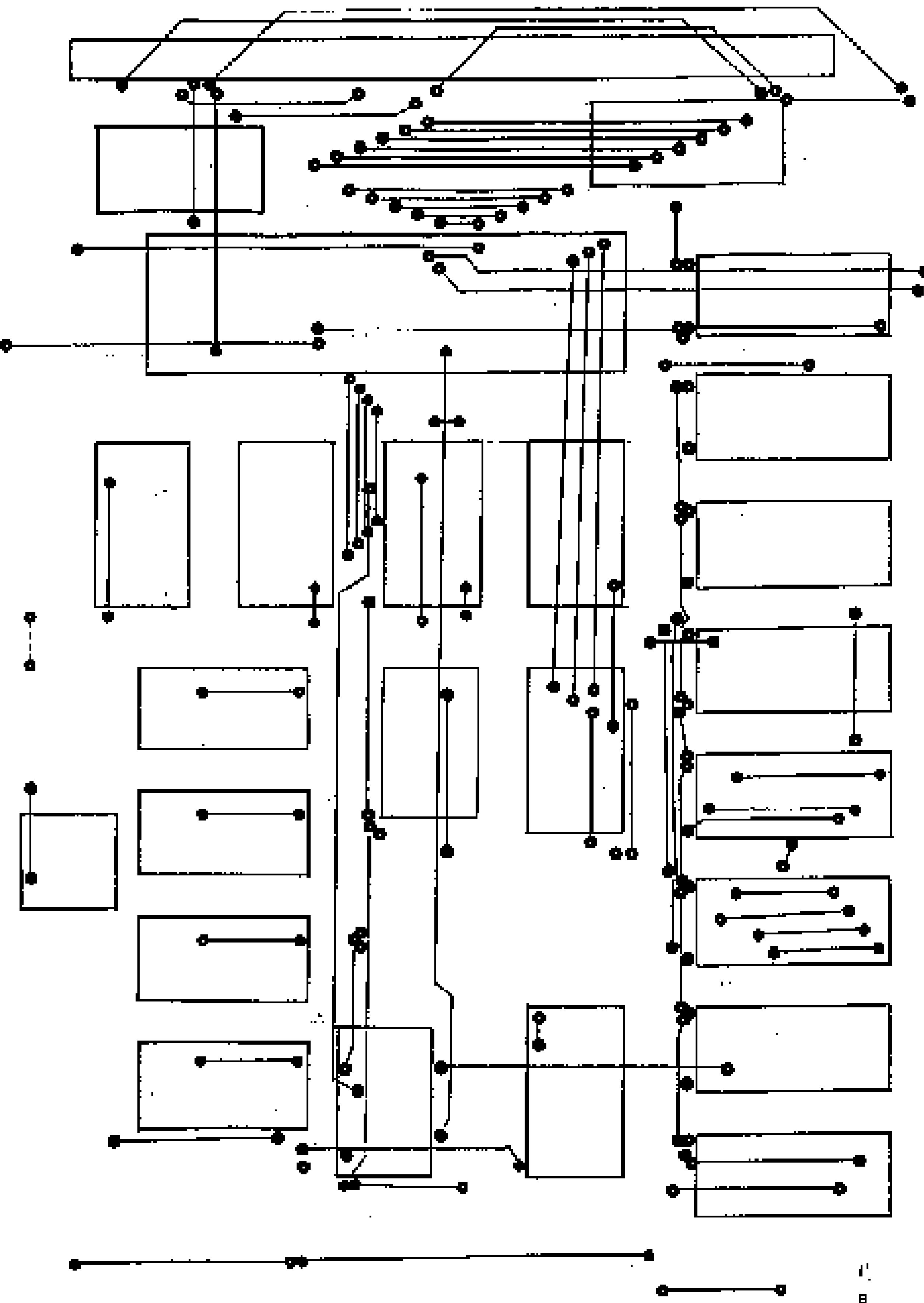


Sorry voor deze onvolkomenheid!

Roland.


Digitizer





```
******************************
*  AUTOMATIC FORMAT PROGRAM  *
******************************
```

door Roland Leurs.

Als je veel diskettes in de aanbieding koopt, moet je veel diskettes formatteren. Heb je bovendien dubbele drives dan moet je nog veel meer diskettes formatteren. Dit alles kost zeer veel tijd als je het zo doet:

```
>*ENABLE
>*FORMAT 1
DO YOU REALLY WANT TO FORMAT DRIVE 1 ?Y
```

enz...

Met onderstaand programma en met het formatteerprogramma van Acornsoft dat je bij een diskpack krijgt is een het formatteren een fluitje van een cent. Er worden na het opstarten geen stomme vragen meer gesteld maar er wordt meteen geformatteerd!
Ik had onlangs pas geleden slechts 1 1/2 uur nodig om 100 diskettes te formatteren, en ze waren niet meteen de eerste keer goed geformatteerd!

Hoe werkt het programma? Heel eenvoudig: na RUN krijg je een inleidende tekst en moet je een password invoeren. Dit heb ik gedaan om onbedoeld gebruik een klein beetje tegen te gaan. Het password staat in regel 80.
Vervolgens moet je aan de hand van de vragen die gesteld worden aangeven in welke drives geformatteerd moet worden. Als je slechts een drive gebruikt, bijv. 0, dan wacht het programma na het formatteren enkele seconden.
Druk om te starten dan op de spatiebalk en het enige wat je hoeft te doen is zorgen voor schijven.

Er staan op de regioschijf twee versies van dit programma. FORMv32 is bedoeld voor 32 of 40 kolomsmode, FORMv80 is geschreven voor 80 koloms (club)kaart met VDU3.4 van Peter Vokke. Voor beide programma's is P-Charme nodig.
De bediening van de twee versies is verder gelijk.

Op de regioschijf staan dus twee programma's. Dit zijn de totale programma's, dus inclusief het programma om te formatteren zelf. De textpage wordt bij *RUN FORMv32 automatisch op #2A00 gezet en als P Charme voorradig start het ook nog.

Dit is alles wat ik hierover te vertellen heb, toch nog een hele lap voor zo'n kort programma. Veel succes ermee !

Roland.

Dit is de listing voor de 80 kolomsversie. De listing voor 32 kolomsversie staat niet afgedrukt, maar is wel te vinden op de regioschijf.

```
 10 REM FORMAT
 20
 30 REM Geschreven door Roland Leurs.
 40
 50 DIM P(64),I(64),DD(3)
 60 FOR N=0 TO 3;DD(N)=0;NEXT N
 70 @=0
 80 $P="START" ; REM PASSWORD VOOR BEVEILIGING
 90
100 PRINT $12
110 NEGATIVE;SERVICE "     ***     FORMAT     ***";NEGATIVE
120 PRINT ''"Dit programma is geschreven om grote aantallen"'
130 PRINT "diskettes te formatteren, zonder dat daarvoor"'
140 PRINT "steeds de juiste commando's voor moeten worden"'
150 PRINT "ingetypt."''
160 PRINT "Ter beveiliging voor ongewenst gebruik of voor"'
170 PRINT "gebruik door onbevoegden moet u eerst het juiste"'
180 PRINT "password intypen."'''
190 PRINT "Geef password  ";INPUT $I
200 IF $I=$P THEN GOTO 220
210 PRINT $7"Password verkeerd !"';END
220 PRINT $12
230 FOR N=0 TO 3
240  PRINT "Moet drive "N" geformatteerd worden  (j/n) ?"
250  INKEY T;IF T=CH"J" OR T=CH"j" THEN DD(N)=1
260  PRINT $T'
270 NEXT N
280 K=DD(0)+DD(1)+DD(2)+DD(3)
290 IF K=0 THEN PRINT ';END
300 PRINT ''"Plaats de schijven op tijd in de drive."'
310 PRINT "de rest gaat vanzelf."''
320 PRINT "Druk op spatie om te starten ... "
330 DO INKEY T;UNTIL T=32
340 ON ERROR GOTO 370
350  N=0
360   IF DD(N)=1 THEN ?#EE=N;PRINT''"FORMATTING DRIVE "N';LINK #284E
370  N=N+1;IF N<4 THEN GOTO 360
380  IF K=1 THEN PRINT "Plaats een nieuwe diskette"';PAUSE 400
390 GOTO 350
```

Oei, was ik bijna mijn vriendelijke groeten vergeten. Bij deze dus !

SIE PIE EMMERTJES.

Zoals bekend is bij Z80-CP/M kaart gebruikers moet je om schijven te formatteren voor de standaard ATOM-DOS gebruik maken van de CP/M programma's DCONF en FORMAT. Dit werkt uitstekend, zij het dat er alleen enkelzijdig geformatteerde schijven aangemaakt kunen worden. Aangezien ik al een poos terug mijn, door het vele copieren op regio avonden doldgedraaide, Olivetti loopwerkje had omgeruild voor een uitstekend werkend dubbelzijdig drive-je van twijfelachtige afkomst, wilde ik toch wel graag dubbelzijdig geformatteerde schijven kunnen realiseren.
Na wat gewroet in DCONF bleek zulks zonder veel moeite mogelijk. Hieronder volgt eerst een deel van de tekst die je voorgeschoteld krijgt wanneer je het HELP commando intikt als je DCONF draait.

· HELP ---

Gebruik de cursorbesturingstoetsen (CTRL SDEX) om de cursor te verplaatsen. Tik bij het format van uw keuze een 'C' in. Hiermee activeert u het instellings commando, waarna u de drive in moet voeren die u ingesteld wilt hebben op het format waar de cursor op staat. Indien het niet mogelijk is de aangewezen drive op het aangegeven format in te stellen, volgt een foutmelding.

A - Toevoegen van een format

Met dit commando kunt u het aantal aanwezige formaten uitbreiden. Zet eerst de cursor op het format dat het meest overeenkomt met het nieuw in te voeren format. Dit vermindert het aantal benodigde wijzigingen. Zie ook het EDIT commando.

E Invoeren van een format

Met dit niet van gevaar ontblote commando kunt u een format toevoegen aan de database. Tijdens het invoeren worden geen reeds aanwezige drive instellingen gewijzigd. Gebruik de cursor besturings toetsen op/neer om de cursor over de velden te bewegen. Gebruik de besturingstoetsen links/rechts, delete en CTRL G om een veld in te voeren. Met de Q keert u weer terug naar het hoofd menu. Uitleg over de velden vindt u elders in dit stukje.
Enkele formaten kunnen worden vernietigd wanneer het EDIT commando wordt geactiveerd (SHARP)

Verklaring van de velden:

Disc size: 0-5", 1-8"
Side Switch:
  0=single side
  1=side sw. after..trks
  2=rev.trk switch(NCR)
  3=cylindric
  4=spec.cyl.(Kapyro)
  5=Special (Sharp)
Density: 0=SD,1=DD
Sector size:
  0-128, 1-256
  2-512, 3-1024

inverse: (Only for sectors>=256)
  0=data not inverted
  1=data inverted on floppy
Blockshift:3=1Kb,4=2Kb,5=4Kb blocs
Skewing
  0=hardware skewing
  1=software skewing
  Direct onder 'Skewing' bevindt zich de skew-tabel. Indien hardware skewing,dan bevat deze info voor het format programma. In overige gevallen wordt bepaald op welke wijze CP/M de sectoren bereikt. (skewing is voor optimalisering van de snelheid.)

De verdere inlichtingen uit het programmadeel HELP zijn voor dit gebeuren niet van belang. Wat nu te doen:

Start uw ATOM op in CP/M mode en "RUN" het programma DCONF. Met de cursorbesturingstoetsen stelt u het program in op ATOM en drukt daarna op toets A.
Nu komt het volgende in beeld:

ADD

input name of new format or just return:

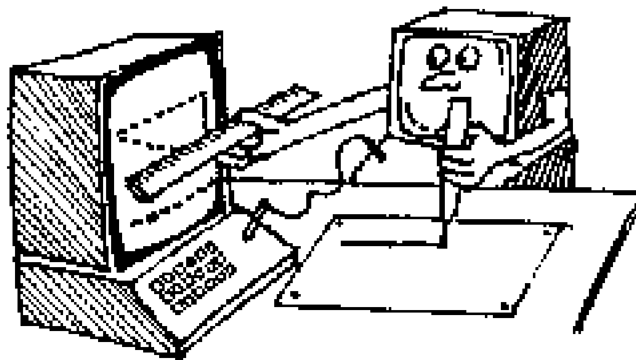
Hierachter vult u in b.v ATOM-DS (ATOM Double Side). Beter niet slechts "ATOM", want dan wordt uw origineie format overschreven. Na de return loopt het scherm vol met de volgende gegevens:

Disc Size	0
Side Switching	0
Cylinders	40
Data density	0
Inverse	0
Sectors per track	10
Sector size	1
First sector	0
Block shift	3
Directory size	64
System tracks	0

Skewing 1
 0 2 4 6

 8 1 3 5

 7 9



De cursor staat te knipperen bij de 0 van Disc size. Met de CTRL C ga je een veld omlaag naar side switching. Tik hier een 1 in en klaar is Kees. Met een Q verlaten we nu dit programmadeel enkomen weer terecht in het hoofd menu. We zien ook dat het nieuwe format keurig in de lijst is opgenomen. Als je nu DCONF netjes verlaat, dus met een X en daarna een Y als antwoord op de vraag of dit format ook in de database moet staat in het vervolg de ATOM-DS steeds voor je klaar.
Als je nu dus de drive instelt op ATOM-DS komen de schijfjes er keurig dubbelzijdig geformatteerd uit. Alleen moet je wel het programma INITDIR op beide kanten van de schijf loslaten. Veel plezier ermee en tot ziens.

Theo Waaijer.

****
* A T O M   M A R K T *
****

TE KOOP AANGEBODEN:

ACORN ATOM met

16 k geheugenkaart
schakelkaart
80 koloms clubkaart
I/O interfase ingebouwd, met o.a. 3x8 LED's
dubbelzijdige TEAC diskdrive
geschakelde voeding
AVT monitor (monochroom met groen beeldscherm)

De moederprint is met alle uitbreidingen in een grote echte pc IBM-kast ingebouwd. Het toetsenbord is op een aparte print los uitgevoerd. Alles is in zeer goede staat en werkt 100% !

Vraagprijs f800,-

Bel voor meer informatie naar Ger Smeitz, 04490-15021.

TE KOOP AANGEBODEN:

1 Acorn Atom voorzien van :
-combikaart
-80 koloms vdu kaart (ACC)
-geschakelde voeding
-1 enkelzijdige disk drive
-monitor (Philips type BM7502 monochroom)
-Diskettebox met alle ACL schijven
 alle jaargangen Atom Nieuws
-zeer veel documentatie
-diverse losse onderdelen (o.a. 2114's, 6502, 8255 etc)
-kleurenkaart
-geheugenkaart
-MDCR met interfase en cassette's

Printer Epson MX80 Graftrax met zeer veel lettertypes

Inlichtingen bij : A.H.C. de Jong
St Josephstraat 82
Sittard
tel 04490-16097

TE KOOP GEVRAAGD:

Een R65C02 of 65C802 processor (eventueel met datasheet)

Peter Wokke

Dukaatstraat 12

1827 CP Alkmaar

Tel: 072 - 621135

```
****************************************
*    AUTOMATISCHE PROGRAMMA SAVER      *
****************************************
```

door Roland Leurs

Iedereen die wel eens programma's intypt (uit Atom Nieuws of eigen programma's) maakt het vroeg of laat wel eens mee dat moeder in de keuken kortsluiting maakt met het strijkijzer. De PLEM heeft daarvoor een beveiliging gemaakt, nl de smeltveiligheid. Bij een kortsluiting brandt dat ding door en de spanning is weg. Maar ook ons programma in de Atom. Wat een ellende.

Met onderstaand programma wordt de schade beperkt tot enkele minuten typewerk. Als we het programma runnen kunnen we een tijdinterval opgeven. Steeds als zo'n periode verstreken is wordt het programma automatisch op schijf gezet, mits men natuurlijk een diskdrive heeft.

Het programma was oorspronkelijk een real time klok op interuptbasis. Het is nu enigzins misvormt en werkt prima. De file wordt weggeschreven vanaf textpage tot de huidige top. Het spreekt voor zich dat er een schijf in de drive moet zitten, anders loopt het hele zaakje vast. De controlleruitbreiding van Sjaak Geene uit AN 7-3 is hier dan ook handig.

Het programma wordt gestopt door even op de breaktoets te drukken of door in te typen ?#238=#00. Hierdoor wordt het juiste tijdinterval nooit bereikt en loopt de interuptroutine wel door, maar er wordt nooit gesaved. Opnieuw opstarten gebeurt dan met het aanlinken van de routine.

Ik heb geprobeerd om deze routine aan te passen voor ED64/ED80 tekstfiles, maar vanwege de werkruimte van de editor en dit programma liep het geheel vast. Ook gaat er nog iets mis met de file die gesaved wordt. Hier hoort U later misschien nog wel meer van.

Ik wens U veel genoegen met dit programma en ik hoop dat we in de redactie nu meer kopy mogen ontvangen; want zeggen dat er geen kopy komt vanwege kortsluiting is vanaf vandaag geen goed excuus meer ...

met de vriendelijke groeten van Roland !

```
   10 PROGRAM AUTOMATIC
   20 REM                    PROGRAM
   30 REM                            SAVER
   40 REM ROLAND LEURS  ACL 1989
   50 DIM SS1,KK6,F7
   60 INPUT "WAAR ASSEMBLEREN "C
   70 P."NA HOEVEEL MINUTEN SAVEN ?";DO INKEY T;U.T>48 AND
T<58;?#80=T;P.$T
   80 INPUT "FILENAAM "$F;IF LEN F>7 THEN F?7=#0D
   90 F.A=1TO6;KKA=#FFF;N.;SS0=#FFF;SS1=#FFF;FOR A=0 TO 1;P.
$21;IF A P.$6
  100 P=C;[
  110:KK0 SEI
  120 LDA@#30;STA#23A;STA#23B
  130 STA#237;STA#238
  140 LDA@KK2%256;STA#204;LDA@KK2/256;STA#205;LDA@13;STA#239
  150 STA#23C;STA#23F;LDA@#40;STA#B80B;LDA@#C0;STA#B80E
  160 LDA@#50;STA#B804;LDA@#C3;STA#B807;STA#B805;CLI;RTS
  170:KK2LDA#B804;INC#236;LDA#236;CMP@20;BNEKK3;LDA@0;STA#236
  180 INC#238;LDA#238;CMP@58;BNEKK3;LDA@48;STA#238
  190 INC#237;LDA#237;CMP@54;BNEKK3;LDA@48;STA#237;
INC#23B;LDA#23B
  200 CMP@58;BNEKK3;LDA@48;STA#23B;INC#23A;LDA#23A;CMP@54;BNEKK3
  210 LDA@48;STA#23A
  220:KK3LDA#235;BEQKK5
  230 BIT#B002;BMIKK3 WAIT FOR FLYBACK
  240 LDA#23B;CMP#80;BNE KK5
  250 LDA@#30;STA #23B;JSR SS0
  260:KK5PLA;RTI
  270:SS0 \ SAVE BASIC PROGRANMA
  280 LDA#CD;PHA;LDX@#C9
  290 LDA@SS1%256;STA #C9;LDA@SS1/256;STA #CA    \ ADRES FILENAAM
  300 LDA@#00;STA #CB;LDA#12;STA #CC      \ BEGIN ADRES = PAGE
  310 LDA@#B2;STA #CD;LDA@#C2;STA #CE     \ EXEC. ADRES = #C2B2
  320 LDA@#00;STA #CF;LDA#12;STA #D0      \ BEGIN ADRES = PAGE
  330 LDA#0D;STA #D1;LDA#0E;STA #D2       \ EIND ADRES  = TOP
  340 LDA@32;STA#AC;JSR #FFDD;LDA@32;STA#AC
  350 PLA;STA #CD;RTS
  360:SS1 \FILENAAM
  370]$P=$F;NEXT;@=0
  380 PRINT "OPSTARTEN MET LINK #"&KK0"',"''
  390 END
```

Wilt u lid worden van de ATOM COMPUTER CLUB?
Neem dan contact op met de penningmeester van de regio waar u bij ingedeeld wilt worden. Deze kan u inlichten omtrent het lidmaatschap.

Regio OVERIJSSEL/GELDERLAND;
Zie regio Arnhem

Regio TWENTE;
G.J.Noorland Prinses Ireneweg 4 7433 DE Schalkhaar
05700-25294

Regio NOORD-HOLLAND;
P. van Kuik Zuideinde 54-a 1843 JP Groot-Schermer
02997-1902

Regio DEN HAAG;
Th.Waayer L.Couperusstraat 6 2274 XP Voorburg
070-862504

Regio DELFT;
Zie regio Den Haag

Regio ROTTERDAM;
Zie regio Den Haag

Regio CENTRUM;
P.van Mourik Ruiterstede 60 3431 XN Nieuwegein
03402-46781

Regio ARNHEM;
J.Hartog Keyenbergseweg 60 6871 WK Renkum
08373-13757

Regio ZEELAND;
E.Gijssel Ruysdaelstraat 6 4462 AD Goes
01100-32557

Regio BRABANT-OOST;
P.Ehrlig Roostenlaan 266 5644 BS Eindhoven
040-114183

Regio LIMBURG;
J.Colen Provincialeweg Z-27 6438 BA Oirsbeek
04492-1957

Regio BELGIE;
Zie Regio Limburg

Bij het aangaan van het lidmaatschap kunt u de contributie overmaken op de rekening van de federatie.Vermeld hierbij uw volledige naam, adres en de regio waar u bij ingedeeld wilt worden.